SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO
Secretaría Académica, de Investigación e Innovación
Dirección de Posgrado, Investigación e Innovación

cenidet
Centro Nacional de Investigación y Desarrollo Tecnológico

Centro Nacional de Investigación y Desarrollo Tecnológico

Subdirección Académica

Departamento de Ciencias Computacionales

TESIS DE MAESTRÍA EN CIENCIAS

Estudio e Implementación de las Mejoras más Relevantes del Algoritmo K-means y su Análisis Comparativo

presentada por
Ing. Jonathan Isai Moreno Cruz

como requisito para la obtención del grado de
Maestro en Ciencias de la Computación

Director de tesis
Dr. Joaquín Pérez Ortega

Cuernavaca, Morelos, México. Octubre de 2016.

Cuernavaca, Morelos a 23 de septiembre del 2016
OFICIO No. DCC/194/2016

Asunto: Aceptación de documento de tesis

C. DR. GERARDO V. GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial del **Ing. Jonathan Isai Moreno Cruz**, con número de control M14CE063, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis profesional titulado **"Estudio e implementación de las mejoras más relevantes del algoritmo k-means y su análisis comparativo"** y hemos encontrado que se han realizado todas las correcciones y observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

DIRECTOR DE TESIS

Dr. Joaquín Pérez Ortega
Doctor en Ciencias Computacionales
4795984

REVISOR 1

Dr. José Crispín Zavala Díaz
Doctor en Ciencias Computacionales
3406871

REVISOR 2

Dr. Moisés González García
Doctor en Ciencias en la Especialidad de Ingeniería Eléctrica
7501724

REVISOR 3

Dra. Alicia Martínez Rebollar
Doctora en Informática
7399055

C.p. M.C. María Elena Gómez Torres - Jefa del Departamento de Servicios Escolares.
Estudiante
Expediente

AMR/lmz

SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO
Centro Nacional de Investigación y Desarrollo Tecnológico

Cuernavaca, Mor., 07 de octubre de 2016
OFICIO No. SAC/299/2016

Asunto: Autorización de impresión de tesis

ING. JONATHAN ISAI MORENO CRUZ
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS
DE LA COMPUTACIÓN
P R E S E N T E

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **"Estudio e implementación de las mejoras más relevantes del algoritmo K-means y su análisis comparativo"**, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

ATENTAMENTE
"CONOCIMIENTO Y TECNOLOGÍA AL SERVICIO DE MÉXICO"

DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO

SEP TecNM
CENTRO NACIONAL
DE INVESTIGACIÓN
Y DESARROLLO
TECNOLÓGICO
SUBDIRECCIÓN
ACADÉMICA

C.p. M.T.I. María Elena Gómez Torres .- Jefa del Departamento de Servicios Escolares.
Expediente

GVGR/mcr

Dedicatoria

"No temas, porque yo estoy contigo; no desmayes, porque yo soy tu ***Dios*** *que te esfuerzo; siempre te ayudaré, siempre te sustentaré con la diestra de mi justicia"*

Isaías 41:10 (RVR60)

Con mucho cariño dedico este trabajo:

A ***Dios***, por permitirme llegar a esta etapa de mi vida, por ser mí ayuda, mi fortaleza y mi refugio en cada momento. *¡****Dios*** *es bueno, siempre bueno!*

A mis padres, ***Santana Moreno*** y ***Araceli Cruz***, por creer en mí, por su amor, su guía y apoyo incondicional, porque son el motor de mi vida.

A mis hermanas, ***Susana*** y ***Stefanny***, por ser las mejores amigas de mi vida, por mostrarme en todo momento su apoyo, consejo y cariño sincero.

A mi sobrino ***Dany***, por traer alegrías a mi vida, por hacerme reír en los momentos más difíciles y darme una motivación para ser mejor persona.

¡Los amo!

Agradecimientos

Mi más profundo agradecimiento al ***Dr. Joaquín Pérez Ortega*** por su apoyo, paciencia, conocimientos y confianza depositada en mí para la realización de este proyecto.

A los miembros del comité revisor: ***Dr. José Crispín Zavala Díaz***, ***Dra. Alicia Martínez Rebollar*** y ***Dr. Moisés González García*** por sus consejos y oportuna opinión científica que ayudaron a fortalecer este trabajo y mi formación profesional.

Al ***Consejo Nacional de Ciencia y Tecnología (CONACYT)*** y al ***Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET)***, por la oportunidad de realizar mis estudios de maestría y brindarme el apoyo y las facilidades durante la realización de esta tesis.

A mi banda, por su amistad y cariño, por permitirme ser parte de ustedes y ser felices en la música. ¡Ahora soy Brunet!

A mis amigos: ***Conny***, ***Genaro***, ***Cesia***, ***Jonh-Rodri***, ***David***, ***Misael***, ***Lupita***, ***Lili***, ***Mau***, ***Viole, Luis y Joe***; por ser mi segunda familia y brindarme su apoyo en todo momento, por sus consejos y por hacer más amena mi estancia en esta ciudad. Los llevaré siempre en mi corazón.

A mis amigos de generación y compañeros de laboratorio: ***Honorio, Tony, Andy, Itzel y Jess***; por todos esos momentos geniales que pasamos juntos, por su amistad y por darle un toque de alegría a cada día.

Reitero mi agradecimiento a ***Dios***, a toda mi ***familia*** y ***amistades*** que estuvieron siempre a mi lado en presencia y a distancia, mostrándome su apoyo y cariño sincero.

¡Gracias!

Resumen

El problema general de la comparación de algoritmos ha sido ampliamente cuestionado, esto, debido a la influencia de enfoques como el teorema *no free luch*, el cual señala que no existe un algoritmo que domine en la solución de todas las instancias de un problema NP. Con este conocimiento, se plantea la cuestión de decidir cuándo un algoritmo es mejor que otro.

Desde la publicación del algoritmo K-means, se han propuesto numerosas mejoras que lo optimizan, sin embargo, no se ha encontrado un mecanismo sistematizado ni herramientas para la comparación de mejoras a K-means en igualdad de condiciones que permitan determinar los casos y características en que una mejora es dominante. En la literatura especializada, existen estudios comparativos del algoritmo K-means, donde, el método clásico de comparación consiste en resolver una instancia de prueba con el algoritmo K-means y la mejora propuesta. Por otra parte, cuando se compara respecto a otras mejoras, se observa la ausencia de elementos importantes que permitan a los investigadores realizar estudios similares, además, existe evidencia de casos en que algunas mejoras se benefician al resolver un determinado tipo de instancia.

En este trabajo, se propone una metodología para la comparación de algoritmos con base experimental y rigor estadístico, la cual, se validó mediante un análisis comparativo de tres de las más relevantes mejoras del algoritmo K-means en su fase de clasificación, a saber: *Early Classification*, *Enhanced K-means* y *Pattern Reduction*. Los resultados obtenidos muestran que en términos de calidad las mejoras dominantes son: *Enhanced K-means* y *Early Classification* con 33 y 28 casos, respectivamente. Por otra parte, en términos de eficiencia, es destacable la superioridad de la mejora *Pattern Reduction*, sin embargo, ésta presentó pérdidas de calidad de hasta 23%.

Esta investigación proporcionará beneficios importantes a los investigadores que requieran comparar diferentes algoritmos heurísticos y a la comunidad científica en general, esto, debido a que los principios de este trabajo pueden aplicarse a otros dominios del conocimiento.

Tabla de contenido

Pág.

Pág.

Lista de figuras

Pág.

Pág.

Lista de tablas

Pág.

Pág.

Capítulo

1

Introducción

"Cuando el objetivo te parezca difícil, no cambies de objetivo; busca un nuevo camino para llegar a él"

Confucio (551-479 a. C.)

En esta investigación se aborda el problema de la selección y comparación de las mejoras más relevantes del algoritmo K-means, las cuales han sido ampliamente utilizadas en diversos dominios.

En las secciones de este capítulo se presenta un panorama general de la tesis, el cual inicia con la contextualización de la investigación y se continúa con la definición del problema. Enseguida, se justifica el desarrollo de esta investigación y se mencionan los objetivos que se plantearon alcanzar. A su vez, se presentan las acotaciones de esta tesis y se explican algunos conceptos relevantes respecto al agrupamiento de datos. En la última sección se presenta una descripción del contenido de cada capítulo de la tesis.

1.1 Contexto de la investigación

El problema general de la comparación de algoritmos se ha estudiado ampliamente en diversos dominios del conocimiento [1]. Uno de los teoremas más influyentes en esta problemática es el teorema *No free luch*, el cual, señala que no existe un algoritmo que domine en la solución de todas las instancias de un problema NP(No Polinomial) [2]. Con este conocimiento, se plantea la cuestión de decidir cuándo un algoritmo es mejor que otro, sin embargo, de acuerdo con McGeoch [3], no existen herramientas analíticas adecuadas para estudiar el desempeño de algoritmos heurísticos; y en los estudios encontrados en la literatura, no se proporcionan las herramientas metodológicas que permitan la sistematización del proceso de comparación de algoritmos.

En la actualidad, existe un creciente interés por resolver problemas del tipo NP, lo cual, ha motivado a los investigadores a proponer enfoques de optimización eficientes y eficaces. En este sentido, desde la publicación del algoritmo K-means [4], se han propuesto numerosas mejoras debido a su elevada complejidad computacional, sobre todo en la solución de instancias de gran tamaño. Adicionalmente, el algoritmo K-means es considerado un problema NP [5], [6], [7] y clasificado como el segundo de diez algoritmos de Minería de Datos más utilizados [8] para la solución de problemas en diferentes dominios [9], [10], [11], [12].

En el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) se ha trabajado desde el año 2005 en la aplicación del algoritmo K-means como técnica de agrupamiento [13], [14], [15],[16]; y se han desarrollado mejoras eficientes al algoritmo, las cuales se describen a continuación:

a) En la tesis de maestría realizada por Basave [17] se presenta una mejora denominada Stop K-means, la cual establece nuevas condiciones de convergencia para el algoritmo. Éstas consisten en: 1) detener el algoritmo cuando en dos iteraciones sucesivas el valor del error al cuadrado de la última iteración es mayor al valor del error al cuadrado de la iteración anterior; 2) parar el algoritmo cuando en dos iteraciones sucesivas los centroides no cambian.

b) En la tesis de maestría realizada por Moreno [18] se propone una mejora denominada P-means, en la cual se introduce que un objeto sólo puede cambiar de membresía a un grupo vecino adyacente, entre una iteración y otra. Este trabajo tiene como objetivo incrementar la eficiencia del algoritmo K-means reduciendo el número de cálculos de distancias.

c) En [19] se propone una mejora al algoritmo K-means denominada Early Classification, cuyo objetivo es identificar aquellos objetos con baja probabilidad de cambio de grupo para ser excluidos de futuros cálculos y aumentar su eficiencia.

d) Como trabajo más reciente destaca la tesis de maestría realizada por López [20]. En este trabajo se propone una mejora denominada N-means, en la cual se identifica que algunos grupos se estabilizan primero que otros, de modo que un grupo estable es aquel que ya no tiene intercambio de objetos con otros grupos en iteraciones posteriores. La idea es discriminar objetos de futuros cálculos de distancia e incrementar la eficiencia del algoritmo.

Los resultados obtenidos en cada investigación han sido alentadores, lo cual motiva la presente tesis de maestría, con la finalidad de continuar realizando trabajos de investigación en el tema. Para nuestro estudio, se seleccionó una de las mejoras desarrolladas en el CENIDET con el objetivo de comparar su desempeño respecto a otras dos mejoras relevantes encontradas en la literatura especializada.

De manera particular, en esta tesis se nombrará a los algoritmos de optimización o variantes del algoritmo K-means como **mejoras del algoritmo K-means** y en la sección 1.7 se describen algunos conceptos importantes utilizados a lo largo de este documento

1.2 Descripción del problema de investigación

Esta tesis surge de la necesidad de comparar las mejoras del algoritmo K-means desarrolladas en el CENIDET respecto a otras mejoras en la fase de clasificación reportadas en la literatura, esto, permitió observar que no existe un mecanismo sistematizado adaptable a nuestro estudio que posibilite la comparación de algoritmos en igualdad de condiciones, además, de la ausencia de estudios de esta índole.

El método clásico de comparación de algoritmos consiste en resolver una instancia de prueba con el algoritmo K-means y la mejora propuesta, con esto, se busca mostrar la eficiencia de dicha mejora. Por otra parte, en ocasiones, cuando se intenta comparar respecto a otras mejoras, se observa la ausencia de información que permita a los investigadores replicar los experimentos. Otro aspecto que algunas veces dificulta la comparación es que en algunas investigaciones se seleccionan instancias de prueba sesgadas [21].

Es común en el dominio de las matemáticas realizar el enunciado del problema mediante la descripción de los parámetros de entrada y las características de una solución válida [22], [23], [24]. De manera particular, el problema que se aborda en esta investigación es la ausencia de un mecanismo sistematizado para la comparación de mejoras del algoritmo K-means en igualdad de condiciones, el cual, permita determinar los casos y características en que una mejora es dominante. Por otra parte, se aborda la selección de dos mejoras más relevantes al algoritmo K-means en su fase de clasificación y su análisis comparativo con la mejora *Early Classification* desarrollada en el CENIDET. Los grandes retos de esta investigación son: 1) Selección e implementación de las mejoras más relevantes del algoritmo K-means para su análisis comparativo y 2) Diseñar e implementar una metodología para la comparación de mejoras del algoritmo K-means (ver Figura 1.1).

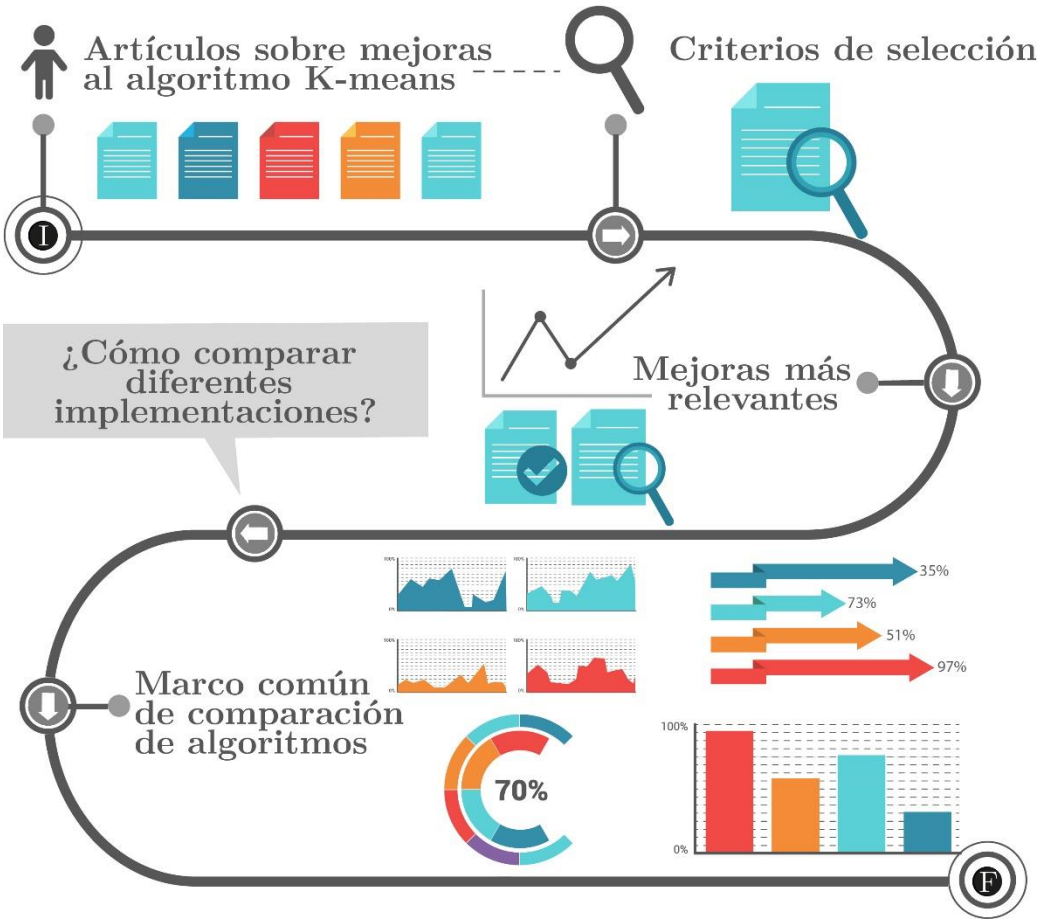


Figura 1.1 Problema de investigación

En la Figura 1.2 se muestra el enunciado del problema de esta investigación. En el segmento a) se muestran los parámetros de entrada con información de las mejoras más relevantes del algoritmo K-means (instancias de prueba, características de los algoritmos e índices de comparación). Es destacable mencionar que en cada publicación se utilizan diferentes instancias de prueba; difieren en la descripción e implementación del algoritmo estándar y la mejora que se propone; y utiliza diferentes índices comparativos, lo cual, limita su comparación respecto a otras mejoras del algoritmo K-means. El segmento b) expresa las características de una solución válida: se tiene un conjunto de instancias de prueba con características comunes para ser probados ampliamente, algoritmos estandarizados y los mismos índices de comparación.

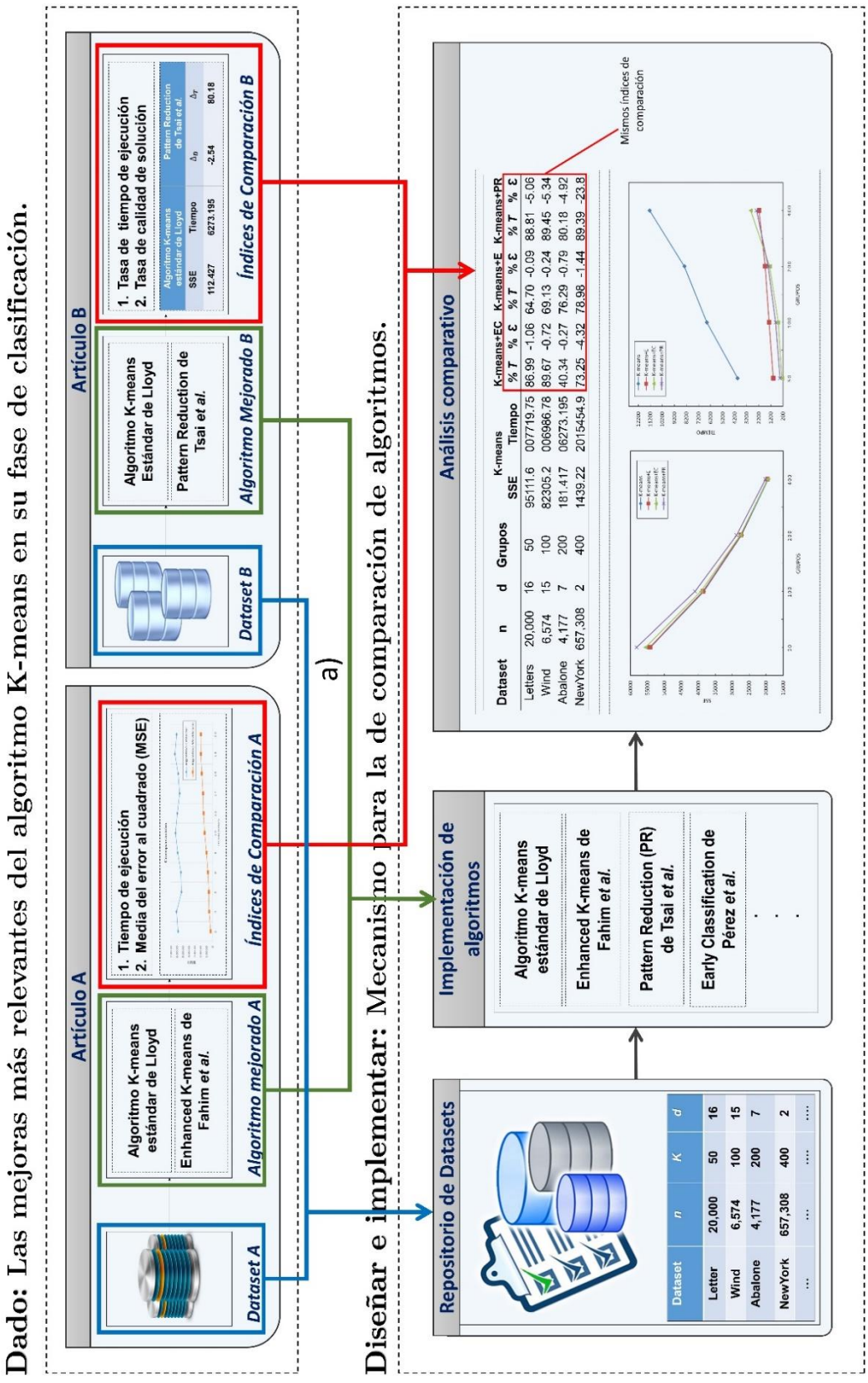


Figura 1.2 Enunciado del problema

1.3 Hipótesis de investigación

Es factible realizar un análisis comparativo de la mejora *Early Classification* y dos mejoras más relevantes del algoritmo K-means en su fase de clasificación mediante métodos experimentales y un mecanismo de comparación de algoritmos.

1.4 Justificación

El desarrollo de esta investigación proporciona beneficios importantes a la comunidad científica, esto, debido a que los principios de este trabajo pueden ser aplicados a otros dominios del conocimiento. Algunos beneficios son:

a) Una metodología detallada para la comparación de algoritmos, la cual aporta rigor estadístico y puede ser aplicada en trabajos futuros de la línea de investigación e incluso en otros dominios de aplicación.
b) Guía práctica para los investigadores que requieran seleccionar y comparar diferentes algoritmos heurísticos para solucionar un problema particular.

1.5 Objetivo

Realizar un análisis comparativo de la mejora *Early Classification* y dos mejoras más relevantes del algoritmo K-means en su fase de clasificación mediante métodos experimentales y un mecanismo de comparación de algoritmos.

1.6 Alcances y limitaciones

Alcances

1) El análisis comparativo incluye tres mejoras del algoritmo K-means.
2) Se implementó la mejora *Early Classification* propuesta en CENIDET y se seleccionaron dos mejoras más relevantes en la fase de clasificación reportadas en la literatura.
3) El análisis compara cuantitativamente los resultados respecto a tiempo y calidad de la solución.
4) Se diseñó un mecanismo para la comparación de mejoras del algoritmo K-means.
5) Con esta investigación se obtuvo un análisis que compara cuantitativamente las mejoras más relevantes del algoritmo K-means en su fase de clasificación.

Limitaciones

1) Para propósitos del análisis comparativo, sólo se seleccionaron aquellos algoritmos cuya información publicada es suficiente para su implementación computacional.
2) Las mejoras del algoritmo se prueban con instancias reales de repositorios reconocidos por la comunidad científica y con instancias sintéticas generadas como parte de la investigación.
3) La implementación de los algoritmos se realizó con el equipo de trabajo y software disponible en el CENIDET.

1.7 Marco conceptual

En esta sección se presentan los tópicos que fundamentan este tema de investigación. Primero, se presenta la notación y los conceptos básicos; posteriormente, se explica el concepto general de agrupamiento de datos y el funcionamiento del algoritmo K-means.

1.7.1 Conceptos básicos

- Objeto: Se refiere a un elemento x tal que $x \in \mathbb{R}^d$, donde $\mathbb{R}^d$ representa el conjunto de números reales en un espacio de d-dimensiones. Otros términos utilizados son: elemento, registro, dato, observación o tupla.

- Dimensiones: Se refiere al número de atributos o características de un objeto y se denota por d.

- Instancia de prueba: Se refiere a una colección de objetos y es denotado por $D = \{x_1, x_2, \dots, x_n\}$, donde n indica el número de objetos. A este término también se hace referencia como conjunto de datos o datos de prueba.

- Grupo: Conjunto de objetos que comparten características similares. El número de grupos se denota por k.

- Centroide: Se refiere al punto medio o patrón representativo de un grupo, el cual es obtenido como la media de todos los objetos pertenecientes a un grupo. Un centroide se denota como m.

- Distancia Euclidiana: En el agrupamiento de datos, la distancia Euclidiana ha sido ampliamente utilizada como una medida de similitud entre dos objetos. Ésta, expresa la distancia entre dos puntos y se puede expresar al menos de dos maneras. La primera, es mediante una función denominada de distancia Euclidiana expresada como $d(x, m)$, donde x y m son los parámetros de la función. En la expresión (1.1), si $x = (x_1, x_2, \dots, x_d)$ y $m = (m_1, m_2, \dots, m_d)$ son dos puntos en un espacio Euclidiano d-dimensional, entonces la función de distancia Euclidiana de un objeto x a un centroide m en un espacio d-dimensional, se expresa como:

$$d(x, m) = \sqrt{\sum_{j=1}^{d} (x_j - m_j)^2} \tag{1.1}$$

Donde x_j y m_j son los valores de la j-ésima dimensión del objeto x y el centroide m, respectivamente.

La posición de un punto en un espacio Euclidiano d-dimensional es un vector cuyos elementos son un conjunto de d números reales, por lo tanto, si x y m son dos vectores, la distancia Euclidiana entre x y m es conocida como el módulo o norma de un vector. En este sentido, la segunda manera de expresar la distancia Euclidiana es mediante la norma Euclidiana denotada como:

$$\|x - m\| = \sqrt{(x_1 - m_1)^2 + (x_2 - m_2)^2 + \cdots + (x_d - m_d)^2} \tag{1.2}$$

Siendo $x = (x_1, x_2, \ldots, x_d)$, donde $x \in \mathbb{R}^d$ y $m = (m_1, m_2, \ldots, m_d)$, donde $m \in \mathbb{R}^d$ dos vectores que representan a un objeto x y un centroide m, respectivamente.

En esta tesis utilizaremos la norma Euclidiana para expresar la distancia Euclidiana de un objeto x a un centroide m.

- Sumatoria del error al cuadrado: Expresa la sumatoria de distancia de los objetos al centroide del grupo al cual pertenece. La expresión 1.3 denota el cálculo de la sumatoria del error al cuadrado y es ampliamente utilizada como referencia de la calidad de la solución.

$$\sum_{i=1}^{k} \sum_{j=1}^{n_i} \|x_{ij} - m_i\|^2 \tag{1.3}$$

Donde k es el número de grupos, n_i es número de objetos en el i-ésimo grupo y $\|x_{ij} - m_i\|^2$ expresa la norma Euclidiana al cuadrado del j-ésimo objeto perteneciente al grupo i al centroide m del grupo i.

1.7.2 Agrupamiento de datos

El proceso de agrupamiento de datos consiste en dividir un conjunto de objetos en subconjuntos llamados grupos, de tal manera que se minimice la sumatoria de las distancias dentro de los grupos y se maximice la distancia entre los centroides de los grupos (ver Figura 1.3), es decir, que los objetos pertenecientes al mismo grupo sean similares entre si y diferentes de los objetos de otros grupos [25].

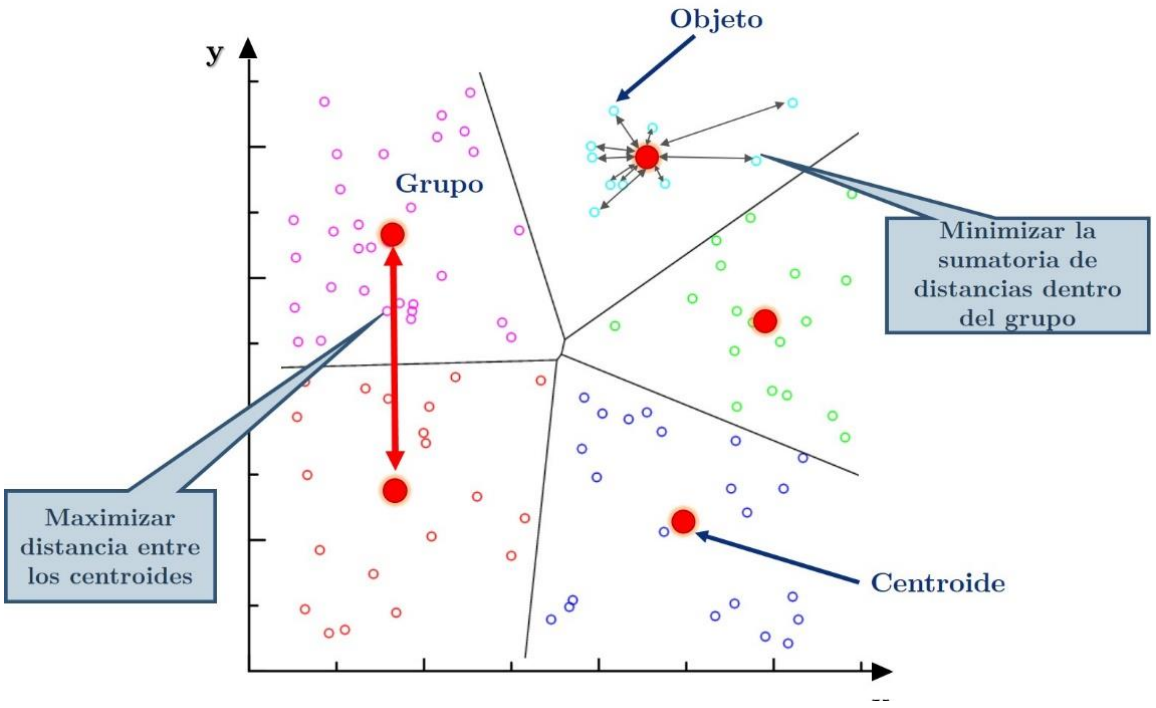


Figura 1.3 Agrupamiento de datos

1.7.3 Algoritmo K-means

El algoritmo K-means es un método iterativo que consiste en particionar un conjunto de n objetos en un número específico de k grupos. Si bien, el término K-means fue acuñado por James MacQueen en 1967 [4], de acuerdo a la literatura especializada en el estudio de los orígenes del algoritmo K-means [6], [26], [27], otros trabajos que describen un proceso similar fueron realizados en 1979 y 1982 por J. A. Hartigan [28] y

Stuart P. Lloyd [29], respectivamente y en la actualidad el más utilizado es el algoritmo de Lloyd.

De acuerdo a la literatura consultada en el proceso de investigación, se observó que la mayoría de los trabajos que reportan mejoras al algoritmo K-means referencian como algoritmo estándar al trabajo realizado por MacQueen, sin embargo, utilizan la idea propuesta por Lloyd. En este contexto, se analizó el procedimiento expuesto por Lloyd con el objetivo de identificar las características del algoritmo.

El algoritmo K-means se ha caracterizado por su amplia utilización en diversos dominios del conocimiento y facilidad de implementación, por lo cual es clasificado como el segundo de los diez algoritmos de agrupamiento más utilizados [8]. Sus parámetros de entrada son un conjunto D con n objetos, el número de k grupos a crear y el número de d-dimensiones, de manera que la complejidad del algoritmo es de orden $O(nkdl)$ donde l indica el número de iteraciones realizadas. El algoritmo se divide en cuatro fases de la forma que se describe a continuación [17], [19]:

1) **Inicialización:** Dados los parámetros iniciales mencionados anteriormente, esta fase consiste en definir el número de k centroides correspondientes a cada grupo. Para este proceso, el método aleatorio es el más comúnmente utilizado, sin embargo, algunas otras implementaciones realizan un pre-procesamiento de los datos para determinar los mejores centroides iniciales.

2) **Clasificación:** En esta fase se realiza el cálculo de distancias de cada objeto a todos los centroides, con el objetivo de identificar y asignar cada objeto al grupo cuyo centroide es el más cercano.

3) **Cálculo de centroides:** Esta fase consiste en el cálculo del nuevo centroide como la media de todos los objetos pertenecientes a cada grupo basándose en la partición actual en cada iteración.

4) **Convergencia:** Esta etapa establece el criterio de paro del algoritmo. Varias condiciones de convergencia han sido utilizadas. Las principales condiciones de convergencia son:

a) Cuando se alcanza un determinado número de iteraciones.
b) Cuando ya no hay cambio de objetos para cada grupo.
c) Cuando en dos iteraciones sucesivas los centroides ya no cambian.
d) Cuando la diferencia de las sumatorias del error al cuadrado en dos iteraciones consecutivas es más pequeña que un umbral dado.

El pseudocódigo del algoritmo K-means se muestra en la Figura 1.4. En la línea 1 se representa la fase de inicialización, de las líneas 3 a 13 se realiza el proceso de clasificación, el cálculo de los nuevos centroides se muestra de la línea 14 a la línea 16 y en la línea 17 se especifica el criterio de convergencia. Si las condiciones de convergencia no son satisfactorias, entonces las etapas 2 y 3 (líneas 3 a 16) se iteran hasta que se cumpla la condición de paro.

```
Algoritmo K-means()
1  Elegir aleatoriamente k objetos del conjunto D como centroides iniciales;
2    Repetir
3      Desde i = 1 hasta n
4        distancia = 0;
5        Desde j = 1 hasta k
6          Calcular la distancia Euclidiana al cuadrado de
           cada objeto a cada centroide dE = ||x_i − m_j||^2;
7          Si distancia > dE Entonces
8            distancia = dE;
9            índice = j;
10         Fin_Si
11       Fin_Desde
12       Asignar el objeto x_i al grupo m_índice;
         grupo_m_índice = grupo_m_índice + x_i;
         n_índice = n_índice + 1;
13     Fin_Desde
14     Desde j = 1 hasta k
15       m_j = grupo_m_índice / n_índice;
16     Fin_Desde
17   Hasta que los centroides no cambien;
```

Figura 1.4 Pseudocódigo del algoritmo K-means estándar

1.8 Organización del documento

El contenido de esta tesis se presenta de la siguiente manera:

En el Capítulo 2 se presenta una descripción de los trabajos consultados referentes a estrategias de comparación y estudios comparativos del algoritmo K-means.

En el Capítulo 3 se propone una metodología para la comparación de algoritmos, la cual, presenta un marco de validación estadística y se describe detalladamente.

En el Capítulo 4 se valida la metodología propuesta y se presentan los resultados obtenidos en el análisis comparativo de mejoras relevantes del algoritmo K-means.

El Capítulo 5 presenta las conclusiones a las que se llegaron en el desarrollo de esta investigación y se concluye con la propuesta de trabajos futuros.

Capítulo

2

Revisión del estado del arte

"Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber"

Albert Einstein (1879-1955)

El problema general del agrupamiento de datos se aborda desde hace muchos años, no sólo con la finalidad de proponer nuevos enfoques y optimizar algoritmos altamente reconocidos, sino también, con el objetivo de identificar aquellos métodos que presentan mayor efectividad al momento de aplicarlos a problemas reales.

Para propósitos de esta investigación, se analizaron los trabajos que realizan un estudio comparativo sobre el algoritmo K-means y trabajos relevantes que presentan estrategias generales para la comparación de algoritmos, los cuales se describen en el contenido de este capítulo.

2.1 Estudios comparativos de versiones del algoritmo K-means

En esta sección se abordan los trabajos que realizan estudios comparativos, los cuales fueron seleccionados por su relevancia con el tema de investigación. A continuación se señalan las particularidades de dichos trabajos.

a) En [30] se realiza una comparación entre el algoritmo K-means y el algoritmo Subtractive Clustering, con la finalidad de identificar ruido en imágenes. Los índices comparativos implementados son la sumatoria del error al cuadrado, el tiempo de ejecución y la precisión del agrupamiento. El objetivo que se persigue es proporcionar recomendaciones a los profesionales y determinar conclusiones importantes adaptables a las aplicaciones *Tracking*.

b) En [31] se propone un análisis comparativo entre dos técnicas de agrupamiento: K-means de Lloyd y K-means Progressive Greedy. En el análisis comparativo se realizó el agrupamiento de datos genéticos mediante cálculos de distancia Euclidiana en un espacio tridimensional bajo las condiciones de separación y homogeneidad, es decir, distancia mayor y menor entre un objeto y su centroide, respectivamente. Como resultado de este trabajo se obtiene un análisis comparativo cuyos índices de comparación son la media de la diferencia al cuadrado (MSD por sus siglas en inglés, Mean Squared-Difference) y el tiempo de ejecución de ambos algoritmos.

c) En [32] se realiza un estudio comparativo entre el algoritmo K-means de MacQeen y el Modelo de Mezclas Normales (MM method), donde, el índice de comparación es la tasa de error de clasificación (MCR por sus siglas en inglés, Misclassification Rates) de ambos algoritmos. Para este fin, se realiza una estimación de la tasa de error de clasificación mediante el teorema de Bayes y se registran los resultados en un Benchmark para su comparación.

d) En [12] se realiza un análisis comparativo entre los algoritmos K-means de Llyod y K-means de Hartigan. La idea es generalizar el algoritmo de Hartigan para cualquier medida de distorsión que corresponde a una divergencia Bregman con el fin de ser típicamente menos sensible a diferentes inicializaciones de centroides. Para el análisis comparativo, se examinan y validan los resultados experimentales realizando la prueba t-student de significancia estadística, y toma como índice de comparación la calidad de los mínimos locales obtenidos en las ejecuciones de ambos algoritmos.

e) En [27] se presenta un importante análisis comparativo de los algoritmos K-means propuestos por Lloyd, MacQueen y Hartigan. Este trabajo se valida de manera experimental tomando como índice de comparación la calidad del agrupamiento. Se evalúa la solución encontrada en cada algoritmo implementando los índices de Dunn y de Jaccard que son dos métricas de calidad del agrupamiento. El primero, es una técnica de evaluación interna que permite encontrar una agrupación compacta donde la solución con el más alto índice de Dunn es considerada la mejor. El segundo, es una técnica de evaluación externa que calcula la similitud entre la solución encontrada y el punto de referencia en porcentaje de clasificación correcta.

f) En [33] se propone un análisis comparativo, en el cual de manera experimental, se aplicaron las pruebas de Friedman y de Iman & Davenport a ocho métodos de inicialización del algoritmo K-means, esto, para determinar si entre ellos existen diferencias estadísticamente significativas. Los índices de comparación que se utilizan para realizar dicho análisis son la sumatoria del error al cuadrado, el número de iteraciones y el tiempo de ejecución de cada algoritmo. El objetivo de este trabajo es mostrar la eficiencia y eficacia de cada uno de los métodos y ofrecer recomendaciones a los profesionales.

Mediante el análisis de los trabajos descritos anteriormente, en la Tabla 2.1 se muestran las principales características de cada investigación. En la columna 2 se especifican los algoritmos comparados en cada artículo, mientras que en la columna 3 se mencionan los índices comparativos generados por cada trabajo.

Tabla 2.1 Tabla comparativa de los trabajos sobre estudios comparativos

Artículo	**Algoritmos Comparados**	**Índice de Comparación**
Marrón 2007 [30]	* Algoritmo K-means. * Algoritmo Subtractive.	* Sumatoria del error al cuadrado (SSE). * Tiempo de ejecución.
Wilkin 2008 [31].	* Algoritmo K-means de Lloyd. * Algoritmo K-means Progressive Greedy.	* Media de la diferencia al cuadrado (MSD). * Tiempo de ejecución.
Qiu 2010 [32]	*Algoritmo K-means de MacQueen. *Método MM (Normal mixture Model).	* Tasa de error de clasificación.
Slonim 2013 [12]	* Algoritmo K-means de Hartigan. * Algoritmo K-means de Lloyd.	* Calidad de los mínimos locales.
Morissette 2013 [27]	* Algoritmo K-means de Lloyd. * Algoritmo K-means de MacQueen. * Algoritmo K-means de Hartigan.	*Índice Jaccard. *Índice Dunn.
Celebi 2013 [33]	* Método de Forgy. * Método de MacQueen. * Método Maximin. * Método de Bradley and Fayyad. * Método K-means++. * Método Greedy K-means++. * Método Var-Part. * Método PCA-Part.	* Sumatoria del error al cuadrado (SSE). * Número de iteraciones. * Tiempo de ejecución.

2.2 Trabajos sobre estrategias de comparación de algoritmos

A continuación se describen los trabajos que presentan estrategias generales para la comparación de algoritmos.

a) En [34] se presenta una metodología para comparar diferentes métodos de agrupamiento, la cual tiene como base un programa de generación de datos sintéticos y una métrica de evaluación. Cabe mencionar que, la metodología que se propone no se describe a detalle, sin embargo, se identificaron algunos aspectos importantes:
    1) La generación de datos sintéticos basados en el número de objetos, dimensiones, número de grupos, tipo de instancia (numérico o nominal) y rango de datos; los cuales son establecidos por el investigador.
    2) Definición de la medida de similitud y la métrica de evaluación basada en la sumatoria de varianzas y el tiempo de ejecución de los diferentes métodos de agrupamiento.
    3) Proceso de experimentación.

b) En [35] se aborda el problema de la evaluación experimental de métodos heurísticos para la optimización de algoritmos. Debido a que en ocasiones los autores utilizan y presentan datos que benefician a sus investigaciones, este trabajo toma algunas consideraciones importantes, por lo tanto, este documento se centra en las cuestiones metodológicas que deben ser consideradas por los investigadores, tales como:
    1) Fuente y diseño de datos de prueba. En este sentido, se presentan algunas recomendaciones importantes sobre el diseño de los datos, sobre todo al ser generados aleatoriamente.
    2) Medidas de rendimiento de algoritmos. En particular, el autor se enfoca en métricas de calidad y no en métricas de eficiencia.
    3) Proceso experimental.

4) Presentación de resultados. En este proceso, se ejemplifican las malas prácticas y se aportan recomendaciones para ofrecer una correcta presentación de resultados.

Es importante señalar, que en este trabajo no se presenta una metodología, sino se centra en recomendaciones a los investigadores para mejorar el proceso experimental.

c) En [3] se presenta una guía experimental que resuelve las principales preguntas, como son: ¿Qué debería medir?, ¿Qué entradas?, ¿Qué es necesario probar?, sólo por mencionar algunas. Este libro se basa en las ideas de diseño de algoritmos, análisis de datos y sistemas informáticos, aportando elementos significativos para el análisis experimental.

Este trabajo, se presenta como una guía en la que se propone una metodología que apoya al proceso experimental y se presentan recomendaciones importantes al momento de diseñar, ejecutar y validar los experimentos. A pesar de no realizar un proceso de comparación detallado, se realizan casos prácticos para ejemplificar cada procedimiento.

d) En [36] se presenta una metodología para el análisis experimental de cuatro algoritmos de agrupamiento. Dicha metodología es particularizada, no se detalla y consta de 7 pasos:

1) Elección de algoritmos de agrupamiento.
2) Selección de instancias de prueba.
3) Importación de datos a la herramienta Weka;
4) Normalización de datos;
5) Ejecución de algoritmos;
6) Almacenamiento de resultados;
7) Graficación

En la Tabla 2.2 se presenta una comparación de los trabajos relacionados a esta tesis.

Tabla 2.2 Tabla comparativa de trabajos relacionados con esta tesis

Característica	Trabajos relacionados										Esta Tesis
	Zaït 1997	**Rardin 2001**	**Marrón 2007**	**Wilkin 2008**	**Qiu 2010**	**McGeoch 2012**	**Slonim 2013**	**Morissette 2013**	**Celebi 2013**	**Bala 2014**	
¿Propone una metodología para la comparación de algoritmos?	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓	✓
¿Se describe detalladamente la metodología propuesta?	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓
¿Proporciona un marco de validación estadística?	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
¿Se enfoca en el algoritmo K-means?	✗	✗	✓	✓	✗	✗	✓	✓	✓	✗	✓
¿Realiza una amplia experimentación?	✓	✗	✓	✗	✗	✗	✓	✗	✓	✗	✓
¿Compara mejoras a K-means en la fase de clasificación?	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
¿Compara al menos 3 algoritmos?	✓	✓	✗	✗	✗	✓	✗	✓	✓	✓	✓
¿Se describen los criterios de selección de algoritmos e instancias de prueba?	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
¿Se describe detalladamente el proceso experimental?	✓	✓	✗	✗	✗	✓	✓	✓	✓	✗	✓
¿Utiliza pruebas estadísticas para validar resultados?	✗	✓	✗	✗	✗	✓	✓	✓	✓	✗	✓
¿Describe los criterios dominación de algoritmos?	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗	✓
¿Evalúa la calidad de agrupamiento?	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓
¿Evalúa el desempeño en tiempo de ejecución?	✓	✗	✓	✓	✗	✓	✗	✗	✓	✓	✓
¿Resuelve instancias sintéticas?	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗	✓
¿Resuelve instancias reales?	✗	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓
¿Resuelve instancias con diferentes niveles de dispersión de datos?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓
¿Se implementaron los algoritmos computacionalmente?	✓	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓

Capítulo

3

Metodología propuesta para el análisis comparativo de algoritmos

Excelente maestro es aquel que, enseñando poco, hace nacer en el alumno un deseo grande de aprender"

Arturo Graf (1848-1913)

La comparación de algoritmos se ha cuestionado y estudiado ampliamente, esto, debido a la ausencia de estrategias adaptables a las necesidades específicas del investigador y que permitan no sólo el proceso experimental, sino también otras especificaciones metodológicas que faciliten el proceso de comparación de manera cuantitativa [3], [37].

En este capítulo se propone una metodología para la comparación de algoritmos, la cual, presenta un marco estadístico y se describe con un mayor nivel de detalle.

3.1 Esquema general de la metodología propuesta para el análisis comparativo de algoritmos

Mediante el análisis de la literatura especializada, se observó que existen estrategias generales para la evaluación de algoritmos [3] [33], [34], [35], [36], sin embargo, no se encontró una metodología adaptable a las necesidades de nuestro estudio (ver Tabla 2.2) y que aporte las especificaciones particulares para el proceso de comparación. Ante esta limitante, se propone una metodología con base en [3], que nos permitirá la realización de nuestro análisis comparativo. Dicha metodología aporta un marco de validación estadística y consta de cuatro fases (ver Figura 3.1).

1) La Fase 1 consiste en definir el objetivo del estudio, es decir, determinar los algoritmos a comparar y definir los índices de comparación.
2) En la Fase 2 se presenta el marco de validación estadística, el cual proporciona aspectos importantes para el diseño experimental, selección de instancias y apoya a la selección de pruebas estadísticas para la comparación y validación de los resultados.
3) La Fase 3 consiste en diseñar y ejecutar un conjunto de experimentos bajo consideraciones estadísticas que satisfagan los requerimientos del estudio comparativo.
4) La Fase 4 consiste en la validación experimental y estadística con el objetivo de determinar patrones de interés y proporcionar observaciones relevantes.

En la Figura 3.1 se presenta la metodología propuesta para la comparación de algoritmos, la cual se describe de manera más detallada en la Sección 3.2.

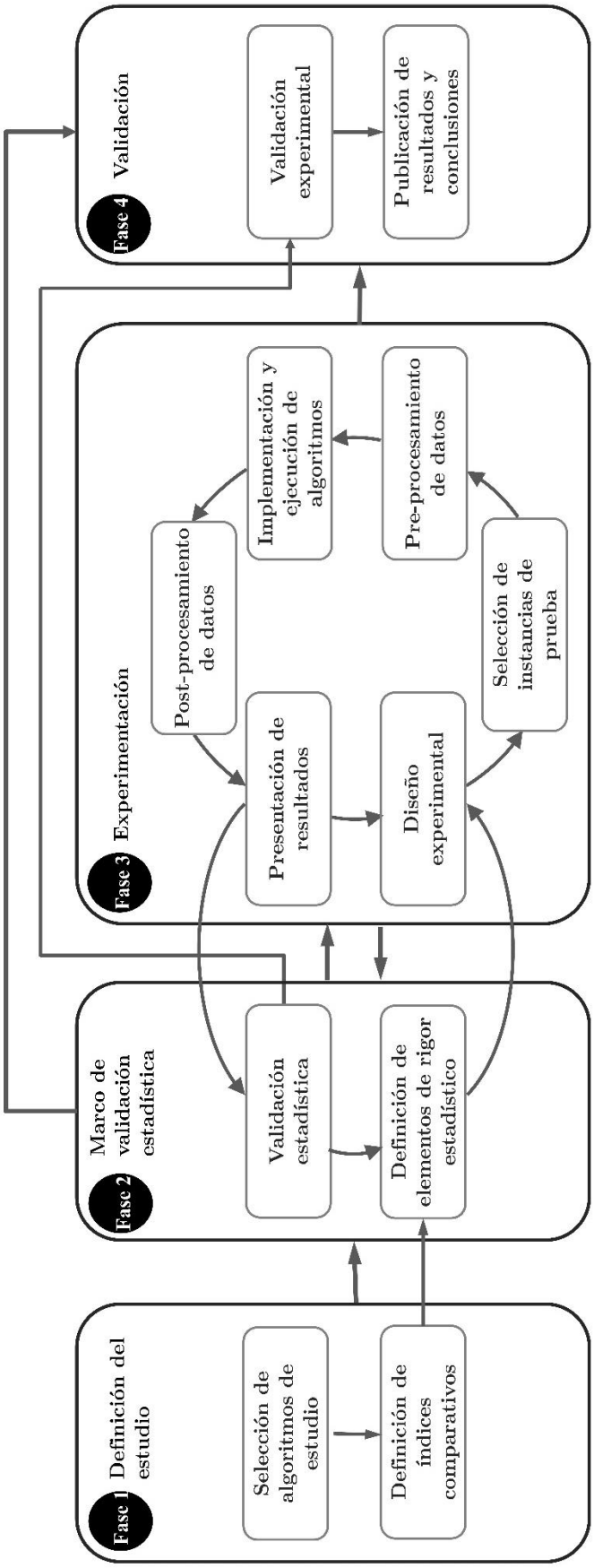


Figura 3.1 Esquema general de la metodología para el análisis comparativo de algoritmos

3.2 Descripción de la metodología propuesta para el análisis comparativo de algoritmos

En esta sección se especifican los procesos correspondientes a cada fase de la metodología propuesta y se presentan aspectos importantes que se deben considerar en el análisis comparativo de algoritmos.

3.2.1 Fase 1: Definición del estudio

El interés por resolver problemas específicos de diversas áreas del conocimiento incrementa el desarrollo de nuevos algoritmos y mejoras que ofrecen métodos eficaces y de interés [1]. Un análisis comparativo cuantitativamente de estos trabajos se convierte en una tarea difícil al ser una gran cantidad de información, sobre todo, si no se cuenta con un objetivo claro de lo que se quiere evaluar. Algunas características que se deben considerar antes de iniciar un análisis comparativo son:

1) Definir el dominio o aplicación del estudio.
2) Delimitar el alcance del estudio.
3) Definir el objetivo, características específicas y condiciones del estudio (¿Qué se quiere medir?, ¿Por qué se quiere medir?, ¿Cómo se va medir?).

Aún con la definición de un dominio en particular, la colección de algoritmos de estudio es elevada. Debido a esto, se presenta una guía para la selección de algoritmos de estudio e índices de comparación que apoyará a los investigadores a realizar este proceso.

a) Selección de algoritmos de estudio

El proceso de selección de algoritmos debe adecuarse a las necesidades y criterios del investigador, sin embargo, tres procedimientos útiles para este proceso se presentan a continuación.

1) La revisión sistemática es un procedimiento de investigación formal que se enfoca en el análisis de estudios científicos originales primarios con el objetivo de buscar información potencialmente relevante y responder a una pregunta de investigación [38]. Dicho procedimiento se resume en 5 pasos: 1) formular una pregunta de investigación; 2) definir criterios de inclusión y exclusión de los estudios; 3) seleccionar estudios relevantes; 4) extraer información relevante de los estudios; 5) analizar y presentar los resultados.

   En la Figura 3.2 se proporciona un conjunto de características que apoyan al proceso de selección de algoritmos de estudio y que el investigador debe considerar con la finalidad de identificar los trabajos relevantes que presenten los elementos necesarios para su implementación computacional.

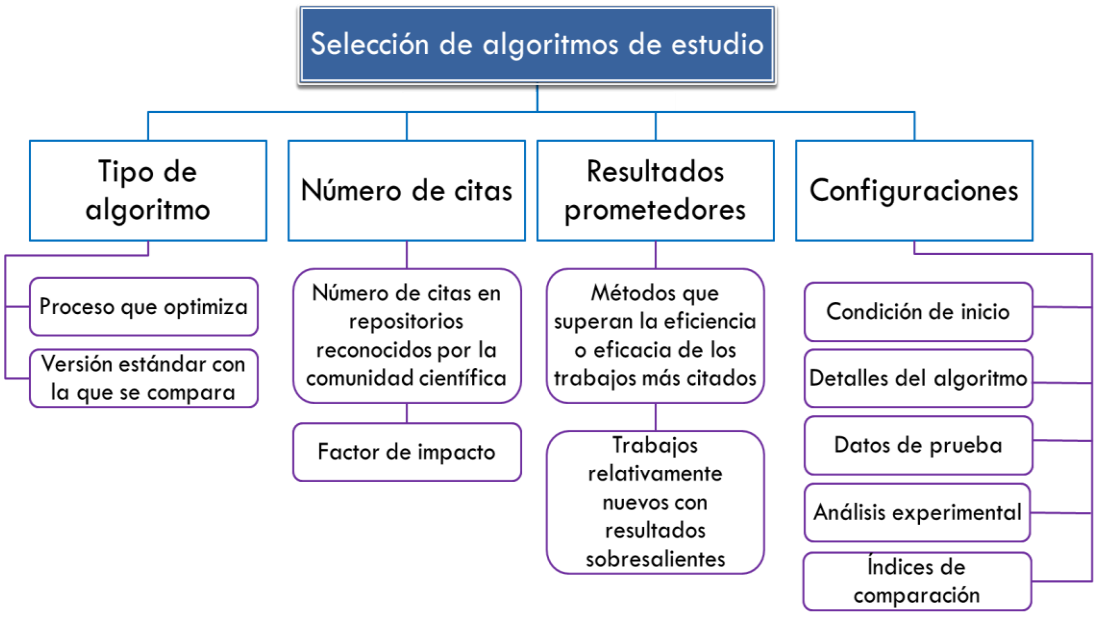


Figura 3.2 Características para el proceso de selección de algoritmos de estudio

2) El muestreo estadístico o probabilístico es un procedimiento que permite elegir un conjunto de elementos representativos de una población [39], es decir, dado un conjunto de artículos que proponen algoritmos, un muestreo nos permite seleccionar un subconjunto representativo de dichos trabajos. Los tipos de muestreo más utilizados son: aleatorizado simple, sistematizado, estratificado y por conglomerados (ir a la Sección 3.2.2), cuya aplicación dependerá del interés del investigador [39], [40], [41].

   Al aplicar este procedimiento podemos inferir sobre una población en particular, sin embargo, es posible que los trabajos seleccionados no correspondan a las características del estudio o no presenten los elementos necesarios para su implementación computacional, lo que implica realizar una detallada definición del estudio y análisis de la muestra en busca de los elementos para su posible reproducción.

3) Una selección a priori es un tipo de muestreo no probabilístico, el cual, se basa en la experiencia, criterios y necesidad del investigador, sin embargo, no es un procedimiento que se recomiende utilizar en todos los casos, ya que limita los alcances del estudio. Este procedimiento puede ser aplicado cuando se quiere evaluar un estudio en particular respecto a otros estudios, para lo cual, se deben especificar los criterios de inclusión y exclusión, así como la justificación de dicha selección.

De manera particular en esta investigación se utilizaron principios de revisión sistemática para la selección de algoritmos, dicho proceso se describe de manera más detallada en la Sección 4.1.

b) Definición de índices comparativos

Las pruebas empíricas y la comparación de algoritmos han sido objeto de discusión e investigación en numerosos contextos, por lo tanto, existen diversas maneras de medir el desempeño, robustez, confiabilidad y precisión de los algoritmos [1], [35].

Los índices comparativos o métricas son elementos que permiten evaluar algoritmos, en particular, en la comparación de algoritmos se deben tomar en cuenta dos factores importantes, la eficiencia y la eficacia, sin embargo, estas métricas se deben elegir de acuerdo al enfoque del estudio. Para esto, en este proceso se plantean dos actividades que apoyarán a la selección de los índices de comparación.

1) Definir los factores de evaluación. Consiste en establecer las características o elementos que se desean comparar de los algoritmos seleccionados.
2) Selección de índices comparativos. Esta actividad consiste en estudiar las métricas o índices utilizados en trabajos que presentan comparaciones entre métodos similares, esto, con el objetivo de seleccionar o crear índices de comparación de acuerdo a lo que se pretende medir en el estudio comparativo.

3.2.2 Fase 2: Marco de validación estadística

Una investigación bien planificada debe contener una formulación objetiva del proceso experimental y de los resultados [40], de manera particular, en un estudio comparativo se deben tomar en cuenta elementos que ofrezcan poder estadístico [37]. El marco de validación estadística que se propone como parte de la metodología consta de dos procesos importantes, los cuales se describen a continuación.

3.2.2.1 Elementos de rigor estadístico

La estadística nos permite estudiar el comportamiento de las observaciones obtenidas en la solución de un problema específico y considera que la inferencia estadística aporta conclusiones acerca de las características de una población en función de una muestra [42].

a) Población, muestra e individuo

La población o universo es el conjunto de elementos del cual estamos interesados en obtener conclusiones, mientras que, una muestra es un subconjunto de individuos o elementos representativos de una población de la cual conocemos sus medidas descriptivas (media, desviación estándar, entre otras) llamados también estadígrafos o estadísticos, mediante los cuales podemos estimar los parámetros de una población e inferir sobre ella [43].

Una población finita es aquella de la cual se conocen todos los individuos y sus características, mientras que, una población infinita es aquella que contiene un número desconocido de individuos, por ejemplo, el número de hormigas que existen en el mundo. Por otra parte, se considera que una muestra es independiente cuando todos los individuos de dicha muestra son diferentes a los individuos contenidos por otra muestra de la misma población, en caso contrario, se dice que las muestras son dependientes o pareadas.

b) Tamaño de la muestra

El tamaño de una muestra está condicionado a los objetivos, necesidades y recursos disponibles que determinarán el diseño, los datos de prueba, las variables que deben considerarse y todo el proceso experimental del estudio. Sin embargo, si el número de individuos que contiene una muestra no es significativo, se incrementa la probabilidad de errar en los resultados obtenidos, mientras que, si se estudia a más individuos de los necesarios, se estarán derrochando recursos [39].

Existen procedimientos que facilitan la elección del tamaño de una muestra [37], [39], [41], [42], [44], [45]. Ahora bien, resulta complicado poder definir el tamaño mínimo de una muestra que satisfaga todos los casos, ya que esto dependerá de cuanto se desvíe la distribución de los datos de una distribución normal.

De acuerdo con el teorema del límite central, entre mayor sea el tamaño de la muestra, ésta será más aproximada a una distribución normal, incluso si la población no se distribuye normalmente. Sin embargo, la literatura especializada en estadística confirma que los autores consideran como lineamiento que una muestra de tamaño $n \geq 30$ presenta una distribución muestral que se aproxima a una distribución normal [41], [42], [44]. Si la población no se aleja de la distribución normal, se puede utilizar una muestra de tamaño $15 < n \leq 30$, de otra forma el tamaño mínimo de una muestra válida será $n = 30$.

c) Tipos de muestreo

El muestreo es una técnica mediante la cual se obtiene una muestra. Las técnicas no probabilísticas son aquellas que se basan en la experiencia y necesidades del estudio, mientras que, las técnicas probabilísticas se basan en la elección aleatorizada y expresan que todos los individuos de la población tienen la misma probabilidad de pertenecer a la muestra [46]. Las técnicas de muestreo más utilizadas son:

1) Muestreo aleatorizado simple: Es una de las técnicas más utilizadas, la cual, consiste en la selección de los individuos que formarán la muestra a partir de un listado de números aleatorios. En este tipo de muestreo es necesario que los individuos de la población estén identificados, de lo contrario se debe optar por otro tipo de muestreo.

2) Muestreo estratificado: Un estrato es un subgrupo de individuos de una población que comparten características similares, por ejemplo: si se quiere comparar los resultados de un estudio psicológico de hombres y mujeres, la población deberá dividirse en al menos dos estratos: hombres y mujeres, y seleccionar una muestra representativa de cada estrato.

Este tipo de muestreo se utiliza cuando se quiere conservar las características de cada estrato de una población. La selección de la muestra será de manera aleatoria para cada estrato de población.

3) Muestreo sistemático: Este tipo de muestreo se inicia con la identificación de los individuos de la población y se realiza una selección inicial aleatoria de un individuo de dicha población. La selección de los n-1 individuos restantes se hará mediante la estimación de la fracción de muestreo, la cual se define como el cociente del tamaño de la población y el tamaño de la muestra, por ejemplo, si se tiene una población de 120 individuos y se quiere extraer una muestra de 30, entonces seleccionará uno de cada 4 individuos hasta formar la muestra de tamaño $n = 30$.

4) Muestreo por conglomerados: Este tipo de muestreo se utiliza cuando se presenta una población demasiado grande. Consta de dos etapas, la primera es la selección de una muestra de la población con una característica específica, mientras que la segunda consiste en la selección de una muestra a partir de la muestra anterior. El número de grupos o características que se incluirán en la selección de la muestra dependerá de las necesidades del estudio. Por ejemplo, si se desea evaluar el nivel de aprovechamiento académico de la educación primaria en escuelas públicas del estado de Oaxaca, primero se deberá seleccionar una muestra de municipios del estado de Oaxaca que cuenten con escuelas públicas a nivel primaria, a partir de la cual, se obtendrá una muestra de alumnos (niñas y niños) sobre la cual se realizará el estudio.

3.2.2.2 Validación estadística

El análisis confirmatorio de datos es un procedimiento que consiste en el uso de pruebas estadísticas que permitan aceptar o rechazar una hipótesis de investigación [42], [43], [47]. En este sentido, a continuación se presentan las bases de la prueba de hipótesis y un conjunto de características que nos permitirán seleccionar adecuadamente una prueba estadística para el contraste de hipótesis.

a) Prueba de hipótesis

Se conoce como hipótesis estadística a un supuesto sobre los parámetros de una población, por lo tanto, la prueba de hipótesis es una prueba estadística que se utiliza para determinar si existe suficiente evidencia para inferir que cierta condición es válida para toda la población [42], [43], [48], en este proceso se examinan dos hipótesis opuestas:

1) Hipótesis nula (H_0): es la afirmación sobre una o más características de la población, que al inicio se supone cierta, es decir, la creencia a priori.
2) Hipótesis alternativa (H_1): es la afirmación contradictoria de H_0, esta hipótesis es la que se espera probar como cierta de acuerdo a los intereses del investigador.

Existen dos tipos de prueba de hipótesis, direccionales y no direccionales. La primera, también llamada prueba unilateral (de un extremo o de una cola), determina si la media de una muestra es menor o mayor con respecto a otra. La segunda, también llamada prueba bilateral (de dos extremos o dos colas), asume la igualdad de medias de las muestras. Ninguna prueba de hipótesis es 100% cierta, puesto que la prueba se basa en probabilidades, siempre existe la posibilidad de obtener una conclusión incorrecta [6]; estos errores son:

1) Error tipo I: También es conocido como α o nivel de significancia. Se define como el rechazo de la hipótesis nula H_0 cuando ésta es verdadera.
2) Error tipo II: También conocido como β, se define como la aceptación de la hipótesis nula H_0 cuando ésta es falsa.

Al contrastar una hipótesis, la máxima probabilidad de cometer el error de tipo I, se llama nivel de significación (α). Cuando aceptamos la hipótesis nula siendo verdadera, tenemos una probabilidad igual a 1-α también conocido como nivel de confianza, mientras que, la probabilidad de rechazar la hipótesis nula cuando es falsa es igual a 1-β, lo cual se conoce como la potencia de la prueba [40], [47]. Para reducir el riesgo de cometer el error de tipo I se debe minimizar el porcentaje de error aceptable, los más utilizados son $\alpha = 0.05$ y $\alpha = 0.01$; en cuanto al error de tipo II, se puede reducir el riesgo de cometerlo si se aumenta el tamaño de la muestra [39], [49].

Una prueba de hipótesis nos conduce a obtener el *p-valor*, el cual es el valor de significancia más pequeño que nos permite rechazar la hipótesis nula. Si el *p-valor* es menor o igual al nivel de significancia, se puede rechazar la hipótesis nula.

b) Selección de pruebas estadísticas

La selección de una prueba estadística depende de diversos factores, los más comúnmente considerados son: 1) el diseño experimental realizado en la investigación; 2) el objetivo o interés del estudio; 3) la distribución de los datos; 4) las preguntas e hipótesis de investigación; 5) la potencia y eficacia de la prueba estadística seleccionada, también conocido como el cumplimento de los supuestos de una prueba [40]. Existen diferentes tipos de pruebas para realizar el contraste de hipótesis y se dividen en dos grandes grupos, pruebas paramétricas y no paramétricas.

Antes de seleccionar cualquier tipo de prueba estadística, se deben considerar 3 supuestos importantes:

1) **Independencia de datos**: Los datos de una muestra deben ser independientes a los datos contenidos en otra muestra, para garantizar este supuesto se recomienda realizar una correcta selección de las muestras. Es importante mencionar, que este supuesto depende de las condiciones experimentales y existen pruebas estadísticas que se adaptan al cumplimiento de este supuesto.

2) **Normalidad de datos:** Hace referencia a la distribución de los datos de las muestras, las cuales deben cumplir con una distribución normal o también conocida gráficamente como la campana de Gauss. Para probar este supuesto existen dos métodos, el primero consiste en generar un histograma de frecuencias, mientras que el segundo consiste en el uso de pruebas de normalidad que determina si la distribución de datos cumple esta condición.

   Un histograma de frecuencia es una representación gráfica de una distribución de frecuencias absolutas o relativas, la cual se utiliza para verificar que los datos cumplan con una distribución normal o campana de Gauss [43]. Esta técnica apoya a la toma de decisiones del investigador acerca del cumplimiento de este supuesto, sin embargo, en la práctica se utilizan pruebas estadísticas que permiten determinar esta condición con mayor exactitud.

   Generalmente, las pruebas de normalidad se aplican mediante paquetes estadísticos y dependen del tamaño de la muestra. Cuando la muestra es $n \leq 30$ se aplica la prueba de Shapiro Wilk, la cual consiste en obtener el *p-valor* de acuerdo a las diferencias entre la simetría de los datos y la forma Gaussiana; mientras que, cuando la muestra es mayor a 30 se utiliza la prueba de Kolmogorov Smirnov,

en la cual se obtiene el *p-valor* mediante la diferencia de la distribución acumulada de los datos de la muestra con la distribución acumulada esperada de una distribución Gaussiana. En ambas pruebas, si el *p-valor* es menor al nivel de significancia (α) elegido, se concluye que los datos no cumplen el supuesto de normalidad [48], [50].

3) **Homocedasticidad de varianzas:** Este término hace referencia a la dispersión de los datos, es decir que las varianzas de las muestras deben ser iguales o similares. Para probar este supuesto se utiliza la prueba de Levene, la cual obtiene el *p-valor* mediante cálculos de varianzas de las muestras. Al igual que en la prueba de normalidad, se rechaza el supuesto, si el *p-valor* es menor al nivel de significancia (α) establecido [40], [42], [48], [50].

En la Figura 3.3 se presenta un diagrama que apoya al proceso de validación del cumplimiento de los supuestos con el objetivo de determinar qué tipo de prueba estadística se debe seleccionar. Para la selección de pruebas paramétricas se debe cumplir obligatoriamente el supuesto de normalidad de datos, mientras que, respecto al supuesto de homocedasticidad de varianzas existen pruebas robustas como ANOVA y t-student, las cuales no incrementan el error tipo I si el supuesto de homocedasticidad no se cumple. Por otra parte, existen pruebas paramétricas y no paramétricas que pueden ser aplicadas cuando se tienen muestras pareadas e independientes.

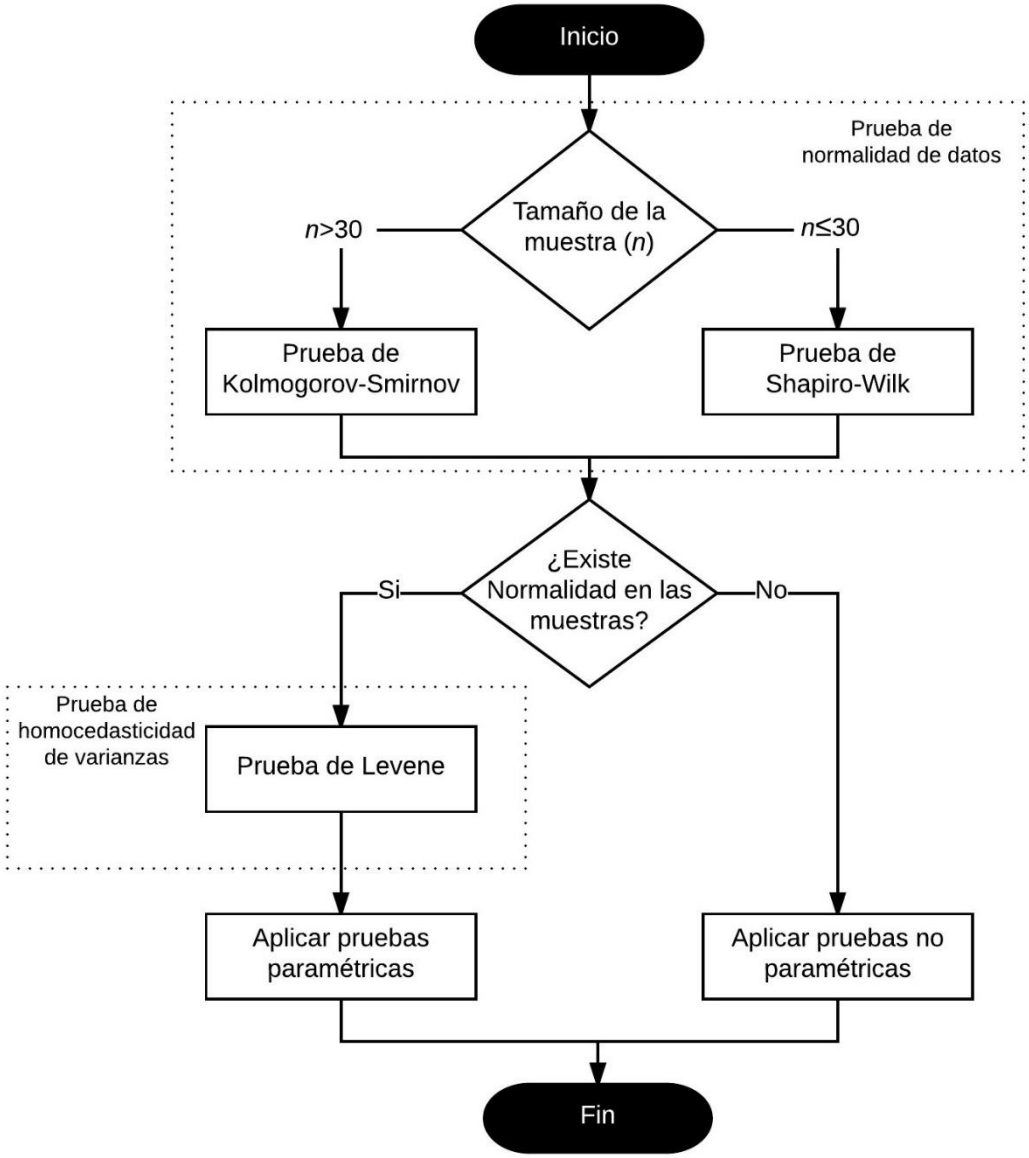


Figura 3.3 Proceso de validación del cumplimiento de los supuestos paramétricos

Una vez evaluados los supuestos paramétricos se debe determinar que prueba paramétrica o no paramétrica se debe utilizar. A manera de resumen en la Tabla 3.1 se concentran las características principales de cada prueba.

Tabla 3.1 Pruebas estadísticas para contraste de hipótesis

Prueba de Hipótesis	**Estadístico de prueba**	**Aplicación**
Prueba t-student [42], [48], [51]	Valor t (diferencia entre estadígrafo y parámetro en unidades de error estándar).	Comparación entre dos muestras de tamaño $n \leq 30$ y una variable dependiente.
Prueba Z [42]	Valor Z (diferencia entre estadígrafo y parámetro en unidades de desviación estándar).	Comparación entre dos muestras de tamaño $n > 30$ y una variable dependiente a la vez.
ANOVA [42], [48], [51]	Valor F (relación de los estimadores de la varianza entre las muestras y dentro de las muestras).	Comparación entre más de dos muestras y una o más variables dependientes.
Wilcoxon [42], [48], [50]–[52]	Valor W_s (valor mínimo de la suma obtenida entre rangos positivos y negativos).	Comparación entre dos muestras pareadas, en la cual si $n \leq 30$ se contrasta con la tabla de Wilcoxon, de lo contrario se estima el valor Z.
U de Mann-Whitney [48], [51]	Valor U (valor mínimo de la suma de rangos de las dos muestras comparadas).	Comparación entre dos muestras independientes en la cual si $n \leq 10$ se contrasta con la tabla de Maan-Whitney de lo contrario es estima el valor Z.
Kruskal-Wallis [42], [48], [51]	Valor K (estadístico obtenido en función de la suma de rangos y los tamaños de las muestras).	Comparación entre más de dos muestras independientes, en la cual si $n \leq 5$ se compara con la tabla de valores críticos mediante la distribución chi cuadrada, de lo contrario se hace la aproximación de dicha distribución.
Friedman [42], [48], [51]	Valor F_R (valor obtenido en función de la suma de rangos y los tamaños muestrales).	Comparación de más de dos muestras pareadas, en la cual si $n \leq 5$ se compara con la tabla de valores críticos mediante la distribución chi cuadrada, de lo contrario se hace la aproximación de dicha distribución.

La selección de pruebas estadísticas para el contraste de hipótesis dependerá de un conjunto de características que deben ser evaluadas, para esto en las Figuras 3.4 y 3.5 se presentan diagramas que apoyarán a la selección de pruebas paramétricas y no paramétricas, respectivamente. A diferencia de otros trabajos, el marco de validación estadística que se presenta, propone el uso de pruebas a posteriori o también llamadas comparaciones múltiples que apoyan a

la medición del tamaño del efecto para determinar que tanto difieren las muestras que se comparan [33], [49], [53]. Por último, se determinan algunas conclusiones importantes en cada prueba estadística, sin embargo, las conclusiones dependerán del objetivo del estudio y de las hipótesis de investigación que se planteen.

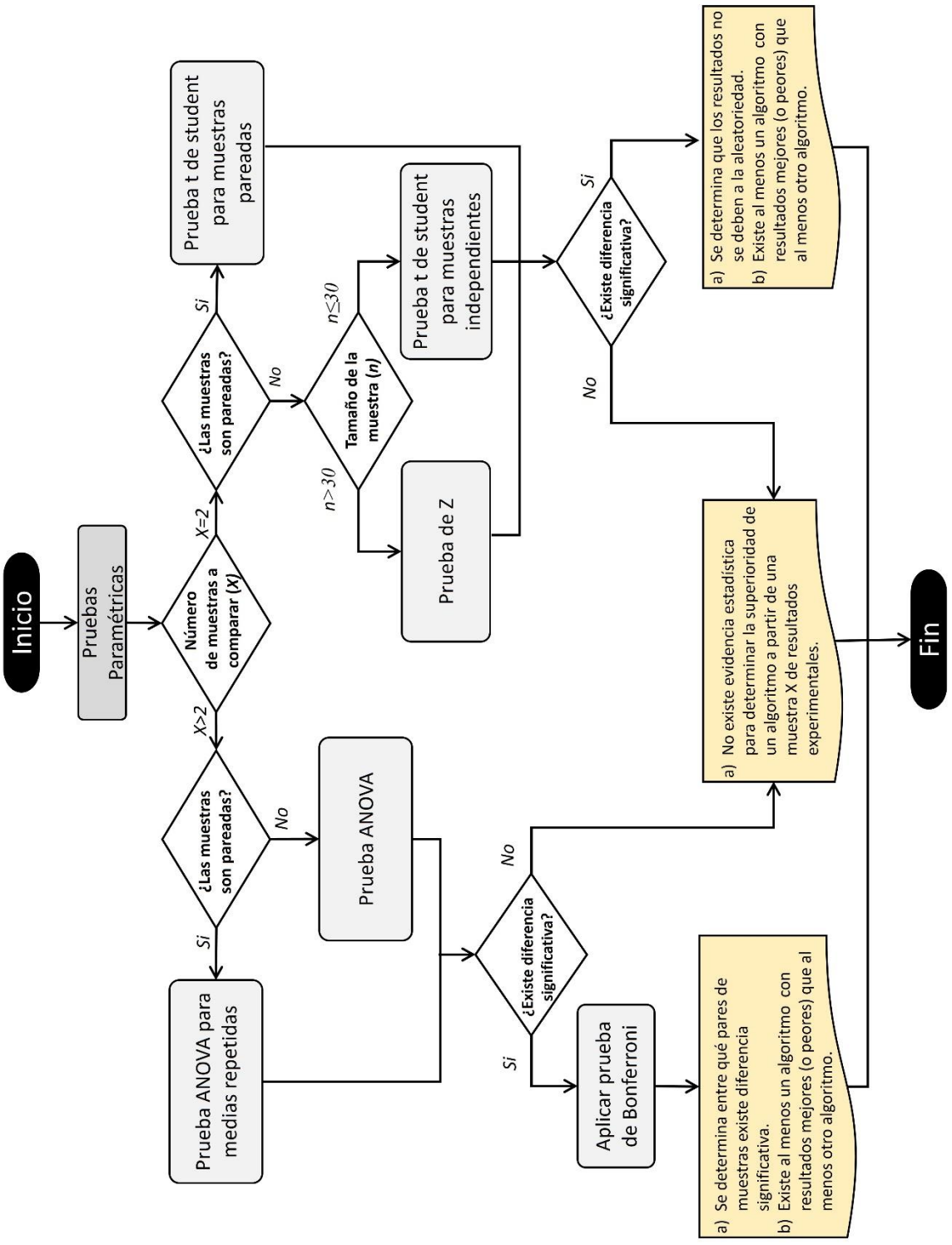


Figura 3.4 Selección de pruebas paramétricas

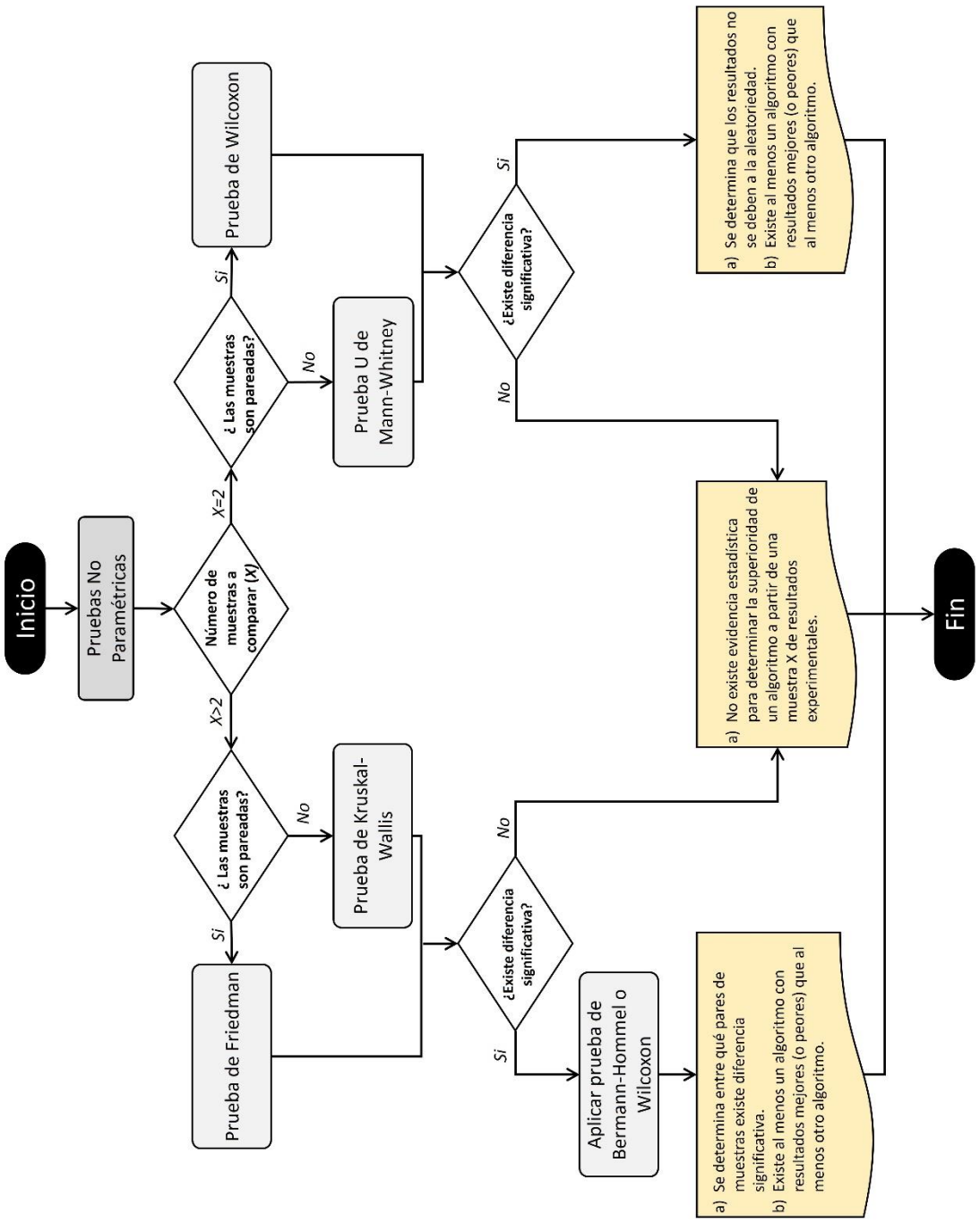


Figura 3.5 Selección de pruebas no paramétricas

3.2.3 Fase 3: Experimentación

La fase de experimentación comprende un conjunto de procesos con fundamento empírico y rigor estadístico, los cuales se describen a continuación.

a) Diseño de experimentos

Este proceso consiste en definir un plan de pruebas que satisfaga las condiciones del estudio y consta de dos pasos:

1) Especificar el entorno de pruebas y las configuraciones para la ejecución de los experimentos.
2) Diseñar un conjunto de experimentos que permita evaluar los algoritmos que se desea comparar tomando algunas consideraciones importantes tales como, 1) objetivo del estudio; 2) ¿Qué características se desean evaluar?; 3) ¿Cómo se medirá?; 4) ¿Qué instancias de prueba se deben seleccionar?; 5) ¿Bajo qué escenarios se ejecutarán los algoritmos?

Es importante señalar que se debe tener una descripción detallada del diseño experimental, de tal manera que se especifique el objetivo de cada prueba, los elementos fundamentales a evaluar y las instancias de prueba para las mismas.

b) Selección de instancias de prueba

Las instancias de prueba son un factor importante en el proceso de comparación y pruebas, ya que existen casos en los cuales algunos autores utilizan datos de prueba que benefician sus resultados [21], y algunos algoritmos son sensibles a los datos de entrada [54]. En este contexto, se sugiere construir un Benchmark con instancias de prueba que satisfagan los criterios y necesidades del estudio comparativo.

Una forma de seleccionar las instancias es mediante un muestreo probabilístico (ir a la Sección 3.2.2.), sin embargo, si se desea evaluar características específicas de diferentes algoritmos, la selección de instancias de prueba dependerá del interés del investigador, en este caso, se deben describir los criterios de inclusión y exclusión y justificar la utilización de dichas instancias. Algunos casos observados en la selección de instancias son:

1) Selección de instancias propuestas en los trabajos que utilizan los algoritmos definidos como objeto del estudio. Al obtener estos datos, se debe verificar su origen y en ocasiones aplicar técnicas de preparación de datos.
2) Selección de instancias reconocidas por la comunidad científica.
3) Creación de instancias de prueba de acuerdo a las características específicas del estudio comparativo. Cuando se generan instancias sintéticas, los investigadores deben procurar un ambiente controlado para no favorecer sus resultados.

c) Pre-procesamiento de datos

Cuando se obtienen instancias reales, podemos enfrentarnos a diversos factores que afectan la calidad y manejo de nuestros datos. En ocasiones los datos provienen de diferentes fuentes, presentan valores atípicos, datos erróneos, datos faltantes e incluso se obtienen en diferentes formatos. Para esto, se recomienda utilizar técnicas de Minería de Datos que apoyen a este proceso. Algunas técnicas de preparación de datos son:

1) Selección de datos
2) Limpieza de datos
3) Construcción de datos
4) Integración de datos
5) Formateo de datos

d) Implementación y ejecución de algoritmos

Este proceso incluye dos actividades importantes las cuales consideramos como el trabajo más costoso dentro de un proyecto de comparación de algoritmos.

1) Análisis detallado de los algoritmos seleccionados con el objetivo de implementarlos computacionalmente. La codificación de los algoritmos debe ser mediante los mismos criterios y recursos con el objetivo de estandarizarlos y tener igualdad de condiciones al momento de ejecutarlos.
2) Diseñar herramientas para la ejecución de algoritmos bajo los escenarios descritos en el diseño experimental. Con base estadística, se recomienda realizar al menos 30 ejecuciones de cada prueba, con el objetivo de reducir el error experimental debido a la aleatoriedad.

e) Post-procesamiento de datos y presentación de resultados

Con el fin de facilitar el análisis y la comparación de los resultados es importante realizar una buena presentación de los datos [55], para esto, se recomienda aplicar un post-procesamiento de los datos de salida, de tal manera que los resultados puedan ser comprendidos de forma tabular, gráfica y textual. Sin embargo, la transformación de datos dependerá de las necesidades del investigador, los índices de comparación y de las especificaciones del estudio comparativo.

3.2.4 Fase 4: Validación

Antes de la validación experimental, se debe determinar la validez de los resultados experimentales, para esto, se deben aplicar pruebas estadísticas de contraste de hipótesis (ir a la Sección 3.2.2). Las hipótesis principales que se deben evaluar son:

1) ¿Los resultados se deben a la aleatoriedad?
2) ¿Existe diferencia estadísticamente significativa entre los resultados de diferentes algoritmos?

3) ¿Existe evidencia estadística para determinar los casos en que un algoritmo es mejor que otro?

a) Validación Experimental

La validación experimental consiste en el análisis de los resultados experimentales y consta de dos procedimientos:

1) Analizar los resultados de tal forma que los datos obtenidos en el proceso experimental no presenten valores atípicos o determinar observaciones que contribuyan al diseño de nuevos experimentos para enriquecer la comparación de algoritmos.
2) Identificar tendencias o patrones de interés y proporcionar conclusiones importantes de dichas observaciones.

b) Conclusiones

En este proceso se deben puntualizar las observaciones más importantes del análisis comparativo y patrones de interés observados durante este proceso, esto con el fin de identificar las características en las que un algoritmo presenta mejores resultados.

Capítulo

4

Aplicación de la metodología propuesta y resultados

"Todos tenemos sueños. Pero para convertir los sueños en realidad, se necesita una gran cantidad de determinación, dedicación, autodisciplina y esfuerzo"

Jesse Owens (1913-1980)

Con el objetivo de validar la metodología propuesta en el Capítulo 3 de esta tesis, se realizó un caso de estudio, el cual consistió en un análisis comparativo de las mejoras más relevantes del algoritmo K-means en su fase de clasificación. A continuación, se describen las particularidades del estudio comparativo y se presentan los resultados obtenidos mediante el proceso experimental, lo que permitió determinar las características y condiciones en las que un algoritmo es la mejor opción.

Este capítulo se presenta de la siguiente manera. Se inicia con la definición del estudio, donde, se puntualizan los procesos de selección de algoritmos e índices comparativos. Después, se describe el diseño experimental, selección de instancias de prueba y las configuraciones experimentales. Por último, se presentan los resultados y el análisis de los mismos.

4.1 Selección de algoritmos de estudio

Respecto a la selección de las mejoras más relevantes, es importante mencionar que existe una gran cantidad de artículos, tesis y libros sobre el algoritmo K-means, lo cual dificulta dicha selección. Para manejar esta complejidad se utilizaron los principios básicos de la revisión sistemática [38]. En este sentido, para la selección de mejoras relevantes se partió de un conjunto de trabajos que presentan mejoras al algoritmo K-means, el cual fue recopilado de importantes bases de datos de acceso libre y se aplicaron tres criterios generales de selección (ver Figura 4.1).

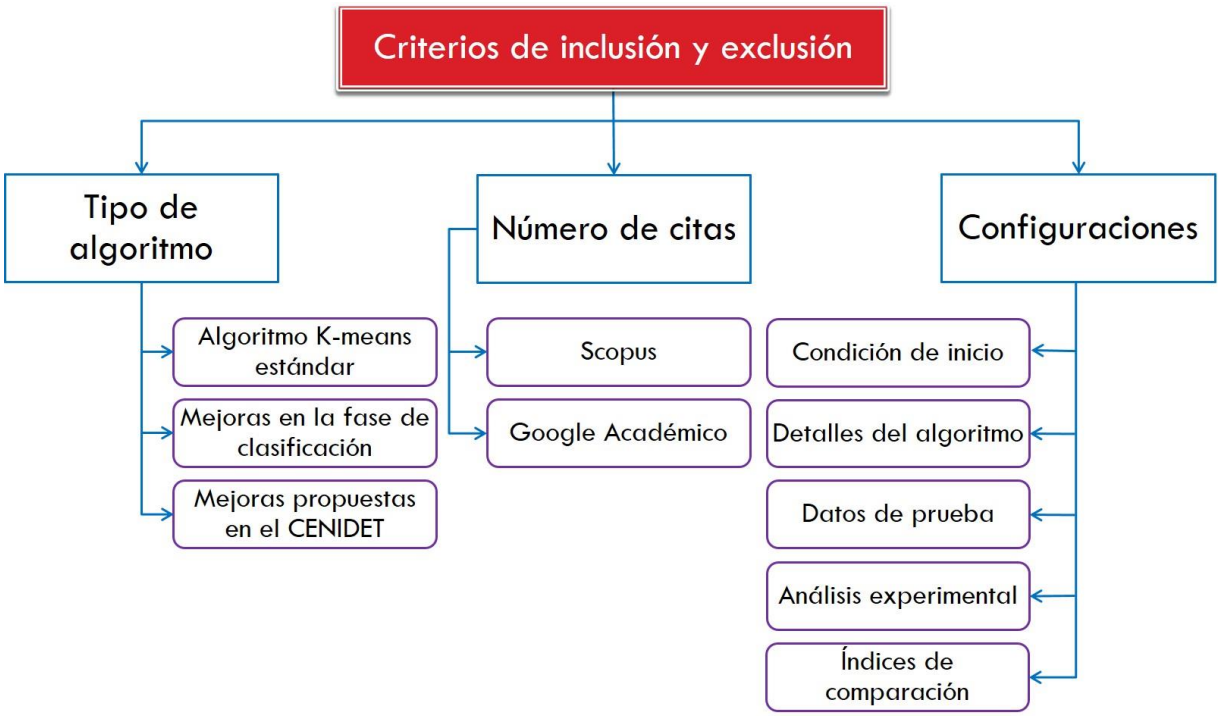


Figura 4.1 Criterios de inclusión y exclusión para la selección de trabajos más relevantes que mejoran el algoritmo K-means

a) Mejoras al algoritmo K-means en su fase de clasificación

De manera general se observó en la literatura que en consenso los investigadores utilizan el algoritmo propuesto por Lloyd como el algoritmo K-means estándar, por esta razón, se eligió dicho trabajo como la versión estándar para la comparación. Como parte de la investigación se determinó

incluir en el análisis comparativo una de las mejoras en la fase de clasificación desarrollada en el CENIDET, por lo tanto, se seleccionó el algoritmo *Early Classification* presentado en [19].

b) Artículos más citados en Scopus y Google Académico.

Respecto al segundo criterio de selección de las dos mejoras restantes para el estudio, se realizó una revisión de 39 trabajos sobre mejoras del algoritmo K-means considerados en esta investigación, esto, con la finalidad de obtener el número de citas en Scopus y Google Académico de cada trabajo. Este filtro permitió realizar un ordenamiento descendiente de los 10 trabajos más citados (ver Tabla 4.1). Es destacable mencionar que se observó la existencia de trabajos recientes que aún no cuentan con número de citas en Scopus y Google Académico, por lo tanto, dichos trabajos no se tomaron en cuenta en nuestro estudio, pero se propone su análisis en trabajos futuros debido a la presentación de resultados prometedores.

Tabla 4.1 Los 10 trabajos más citados que mejoran el algoritmo K-means

ID	Autor	Año	No. de Citas	
			Scopus	Google A.
1	Kanungo [56]	2002	4019	2504
2	Elkan [57]	2003	156	495
3	Fahim [58]	2006	63	193
4	Lai [59]	2009	26	56
5	Lai [60]	2010	17	38
6	Lai [11]	2008	14	30
7	Chiang [61]	2011	12	63
8	Wang [62]	2012	8	28
9	Osamor [63]	2012	2	9
10	Lee [64]	2012	2	7

Los 10 trabajos que proponen mejoras al algoritmo K-means más citados se describen a continuación:

1) En [56] se presenta una mejora al algoritmo K-means basada en una estructura de datos kd-tree, denominado algoritmo de filtrado. La idea es mantener para cada nodo del árbol un subconjunto de centroides vecinos, para los cuales un objeto puede ser movido.

2) En [57] se propone una mejora que evita cálculos de distancia innecesarios aplicando el teorema de la desigualdad triangular, el cual, hace un seguimiento de los límites superior e inferior para las distancias entre los puntos y los centroides.

3) En [58] se presenta una mejora al algoritmo K-means, la cual consiste en calcular y mantener la distancia al centroide más cercano para cada objeto en cada iteración, de esta manera, en la iteración posterior se calcula la distancia del objeto a centroide previamente obtenido. El objeto permanecerá en el mismo grupo si la nueva distancia es menor o igual a la distancia anterior, de otra forma, se calculan las distancias a los otros k-1 grupos.

4) En [11] se realiza una mejora al algoritmo propuesto en [56]. La eficiencia de esta implementación se debe al uso de un árbol binario, el cual, asocia los objetos con cada nodo del árbol y utiliza la información de los desplazamientos de los centroides para determinar el conjunto de vecinos candidatos para cada nodo.

5) En [59] se propone una mejora en la que se calculan los desplazamientos de los centroides entre la iteración previa y la actual. La idea es identificar a los grupos como estáticos o activos, de modo que sólo se calculará la distancia a los centroides activos más cercanos a los objetos y así reducir el tiempo computacional.

6) En [60] se presenta una mejora al algoritmo K-means, la cual hace uso de la información de los puntos y su pertenencia al grupo para determinar el centroide más cercano. Otro factor importante, es la selección del conjunto de centroides iniciales mediante el cálculo de desigualdades con la finalidad de reducir su complejidad computacional en instancias multidimensionales.

7) En [61] se propone una mejora que comprende dos métodos: el primero es utilizado para comprimir y remover objetos que se localizan más cercanos al centroide; el segundo, consiste en asignar los objetos al grupo cuyo centroide es el más cercano y actualizar dicho centroide. La idea radica en comprimir y eliminar objetos poco probables a cambiar de membresía en cada iteración con la finalidad de excluirlos de futuros cálculos y reducir el número de iteraciones realizadas por el algoritmo.

8) En [62] se propone una mejora, donde se identifican los puntos activos y cercanos a los límites de cada grupo, a partir del cual se genera un conjunto de vecinos candidatos para cada objeto utilizando árboles BSP. La idea es minimizar el número de cálculos de distancia y reducir el costo computacional del algoritmo.

9) En [63] se presenta una mejora, en la cual se utiliza una métrica para determinar la estabilidad de un grupo, de modo que si los miembros de un grupo no se mueven en iteraciones posteriores, éstos, tienen poca probabilidad de cambiar de grupo en futuras iteraciones y son candidatos para ser excluidos de cálculos.

10) En [64] se realiza una mejora al algoritmo propuesto en [58], en el que se proponen dos reglas. La regla de selección es usada para adquirir buenos candidatos como centroides iniciales, mientras que la

regla de eliminación consiste en descartar uno o más centroides no calificados para los cálculos de distancia.

c) Análisis de trabajos en busca de elementos para su implementación computacional.

El tercer criterio de selección fue aplicado sobre los trabajos presentados en la Tabla 4.1, esto, con el objetivo de identificar las características que permitieran su reproducción. Mediante este análisis se determinó que los trabajos 2 y 6 propuestos por Elkan [57] y Lai [11], respectivamente, no proporcionan los elementos necesarios para su replicación. De acuerdo con la Tabla 4.1, en los trabajos 1, 4 y 5 propuestos por Kanungo [56] y Lai [59] , [60], respectivamente, se proponen mejoras al algoritmo K-means tomando como base estructuras de datos, tales como árboles binarios y kd-tree, por lo tanto, dichos trabajos fueron descartados como posibles implementaciones debido a que no se ubicaban dentro del objeto de estudio.

Por otra parte, se determinó que los trabajos 3 y 7 propuestos por Fahim [58] y Chiang [61], respectivamente, son los trabajos más citados que cumplen con todos los criterios de selección. Como resultado de este proceso se eligieron 3 mejoras consideradas como más relevantes al algoritmo K-means en su fase de clasificación, a saber: *Early Classification* [19], *Enhanced K-means* [58] y *Pattern Reduction* [61]. Para más detalles de los algoritmos seleccionados, ir al Anexo A.

4.2 Definición de índices comparativos

Mediante la exploración de trabajos relacionados se realizó un listado de los índices comparativos utilizados en cada trabajo, obteniendo un total de 12 métricas de comparación sin repetición. El objetivo fue observar la manera en que otros autores resuelven el problema de la comparación de algoritmos y la forma en que miden sus resultados. Es destacable mencionar que cada autor utiliza diferentes índices comparativos, lo cual, complica el contraste de resultados de diferentes trabajos e incluso determinar en qué circunstancias un algoritmo presenta los mejores resultados.

De acuerdo con [35], los aspectos más importantes a evaluar en la comparación de algoritmos son la eficiencia y eficacia, en este sentido, para este estudio se determinó la evaluación del desempeño de los algoritmos seleccionados mediante métricas de eficiencia y eficacia, esto es, en términos de porcentaje de reducción de tiempo (ver expresión 4.1) y el porcentaje de pérdida de calidad (ver expresión 4.2).

$$\varepsilon = \frac{(SSE_1 - SSE_2) * 100}{SSE_1} \tag{4.1}$$

Donde, SSE_1 expresa la sumatoria del error al cuadrado obtenida por el algoritmo estándar y SSE_2 expresa la sumatoria del error al cuadrado obtenida por el algoritmo mejorado.

$$\tau = \frac{(t_1 - t_2) * 100}{t_1} \tag{4.2}$$

Donde, t_1 expresa el tiempo de ejecución obtenido por el algoritmo estándar y t_2 expresa el tiempo de ejecución obtenido por el algoritmo mejorado, ambos con unidad de tiempo en milisegundos.

4.3 Diseño experimental

Con la finalidad de analizar el desempeño de las mejoras seleccionadas en la Sección 4.1, se diseñó un conjunto de experimentos que permitió observar el comportamiento de las mejoras más relevantes respecto al algoritmo K-means estándar y entre ellas. Es importante destacar dos conjuntos de configuraciones, el primero, expresa los criterios de selección de instancias (ir a Sección 4.4) y el segundo describe las configuraciones computacionales y experimentales (ir a Sección 4.5).

Respecto a la selección de instancias de prueba, no se siguió un muestreo probabilístico, en el caso particular de nuestro estudio, se seleccionaron las instancias de prueba resueltas en las mejoras más relevantes previamente seleccionadas, esto, con la finalidad de replicar sus resultados y comparar su desempeño respecto a las otras mejoras. Para más detalle de las instancias de prueba, ir a la Sección 4.4.

Considerando los parámetros n, k y d de la complejidad del algoritmo K-means, a continuación se describen los experimentos propuestos para el análisis comparativo.

a) El experimento A se realiza bajo dos condiciones:

   1) Resolver las instancias reales R1, R4 y R8 resueltas en el trabajo propuesto por Fahim [58], con incrementos en el número de grupos. Con esto se busca observar el comportamiento de las mejoras del algoritmo K-means y su versión estándar al resolver las instancias con diferentes valores de k. En estos casos, los valores de n y d se mantienen fijos para cada instancia, mientras que el valor de k se incrementa para cada instancia como se describe en la Tabla 4.2 (ir a la página 57).

   2) Resolver las instancias sintéticas S1, S2, S3 y S4 resueltas en el trabajo propuesto por Pérez [19], fijando los valores n y d de acuerdo a las características mencionadas en la Tabla 4.3 (ir a la página 57) e incrementar el número de grupos en k= 50, k=100, k=200, k=400 y k=800. El objetivo es observar la manera en que el algoritmo K-means

estándar y las mejoras relevantes seleccionadas se comportan al realizar incrementos en el número de grupos y que tanto difieren los resultados respecto a los resultados con instancias reales.

b) El experimento B se realiza bajo las siguientes condiciones:

Dado que las instancias sintéticas S1, S2, S3, y S4 son conjuntos de datos enteros uniformemente distribuidos en un rango de valores similares (1-200), se aprovechan las pruebas realizadas en el experimento A con dichas instancias. El objetivo de este experimento es analizar el comportamiento de las mejoras del algoritmo K-means y su versión estándar al realizar incrementos en el número de objetos.

Para estas pruebas se fija el valor d=2 para todas las instancias, y se analizan cinco casos:

1) Cuando k=50 y se incrementa el número de objetos en n=2500, n=10000, n=20000, n=40000.

2) Cuando k=100 y se incrementa el número de objetos en n=2500, n=10000, n=20000, n=40000.

3) Cuando k=200 y se incrementa el número de objetos en n=2500, n=10000, n=20000, n=40000.

4) Cuando k=400 y se incrementa el número de objetos en n=2500, n=10000, n=20000, n=40000.

5) Cuando k=800 y se incrementa el número de objetos en n=2500, n=10000, n=20000, n=40000.

c) El experimento C se caracteriza por resolver las instancias S5, S6, S7, S8, S9, S10, S11 y S12 resueltas en el trabajo propuesto por Chiang [61], con el objetivo de observar el desempeño del algoritmo K-means estándar y las mejoras más relevantes seleccionadas, al incrementar el número de

dimensiones. Para estos casos, se mantienen fijos los valores n=6000 y k=50 para cada instancia, mientras que los valores de d se incrementaron de la siguiente manera; d=2, d=10, d=25, d=50, d=100, d=250, d=500, d=1000.

Adicionalmente, se desea observar el comportamiento de las mejoras seleccionadas y el algoritmo K-means estándar en tres casos particulares:

1) Al resolver la instancia R3 resuelta en [19] con valores de n y d iguales a 150 y 3, respectivamente, e incrementando el número de grupos en k=3, k=5, k=10, k=20, k=30, k=40, k=50. Es importante destacar que R3 es la reconocida instancia Iris de la cual se conoce su agrupamiento óptimo y se busca analizar los casos de coincidencia por cada mejora.

2) Al resolver instancias sintéticas de tipo real creadas aleatoriamente en un rango de valores de 0-1 con una distribución normal y gran cantidad de datos tales como: S13 y S14 resueltas en [61]. La primera, con valores n=60000, d=2 y k=50. La segunda con valores n=600000, d=2 y k=50.

3) Al resolver instancias de prueba reales con una mayor dispersión de datos, tales como R7 resuelta en [61]; y R2, R5 y R6 resueltas en [19]. El objetivo es observar el comportamiento de las mejoras al resolver instancias apegadas a la solución de problemas reales.

4.4 Benchmark de instancias de prueba

Previo a la descripción de los criterios de selección de las instancias de prueba para el análisis comparativo, es importante destacar la existencia de dos tipos de instancias de prueba: sintéticas y reales. Las instancias sintéticas se caracterizan por su creación de acuerdo a las necesidades del estudio, de manera general, se ha caracterizado la creación de instancias sintéticas con distribución normal con el objetivo de tener un ambiente controlado en las prácticas experimentales. Por otra parte, las instancias reales son conjuntos de datos que por lo general presentan mayor dispersión e incluso pueden

alejarse de una distribución normal. Estos datos provienen de diferentes dominios y aplicaciones del mundo real, por ejemplo, datos médicos, fotográficos, biológicos, coordenados, por mencionar sólo algunos. A modo de ejemplo en las Figuras 4.2 y 4.3 se muestran las distribuciones de instancias sintéticas y reales, respectivamente.

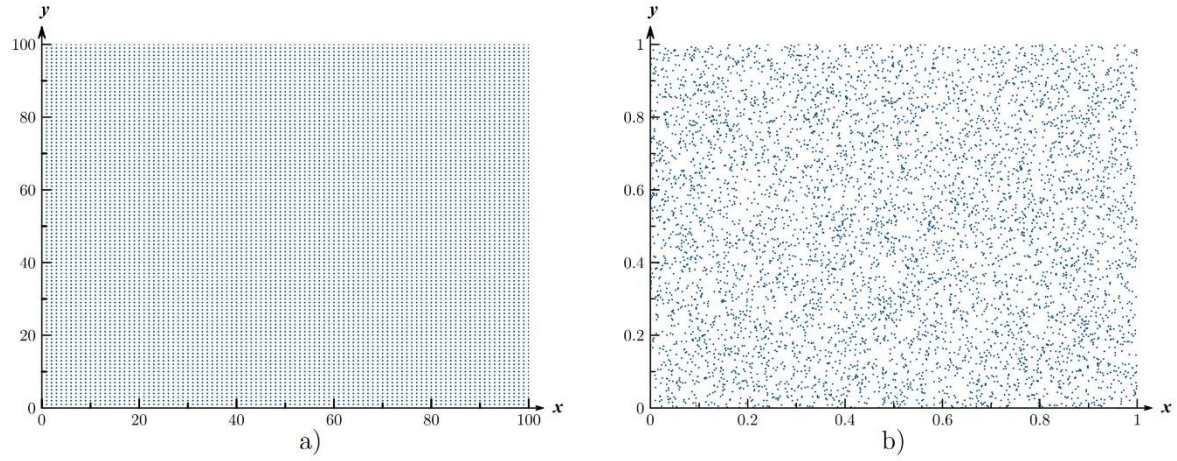


Figura 4.2 Distribución de instancias sintéticas: a) Instancia bidimensional con n=10000 objetos uniformemente distribuidos en un rango de valores de 1-100; b) Instancia bidimensional con n=6000 objetos aleatorios en un rango de valores de 0-1.

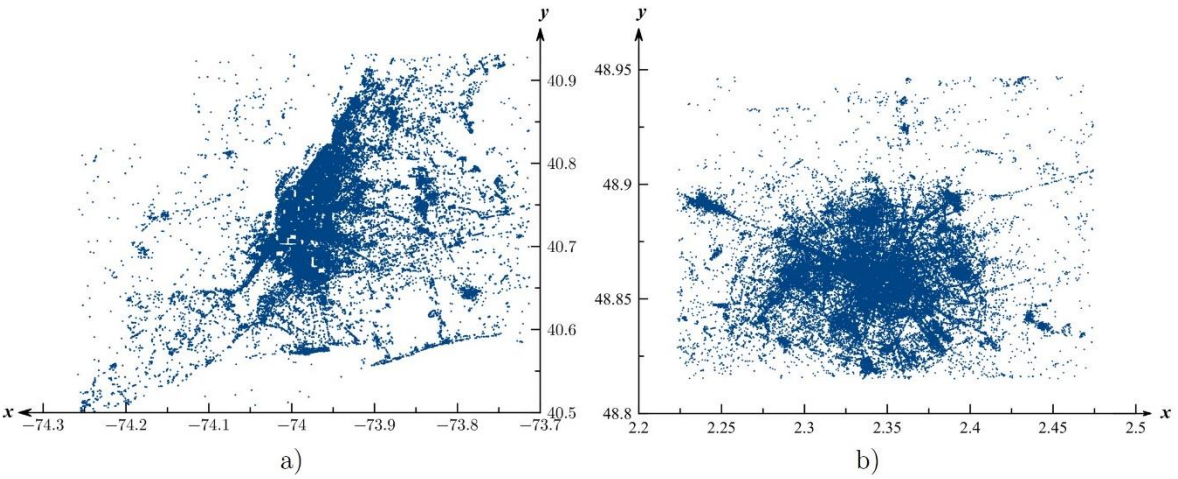


Figura 4.3 Distribución de instancias reales: a) Instancia New York con n=657308 y d=2; b) Instancia París con n= 414528 y d=2.

Mediante el análisis de trabajos del estado del arte y trabajos realizados en CENIDET se obtuvieron 83 instancias de prueba con las cuales se construyó un Benchmark que costa de 30 instancias sintéticas y 53 instancias reales obtenidas de importantes repositorios reconocidos por la comunidad científica [65], [66] y [67].

Durante este proceso se observó que hay instancias que se dividen en varios archivos, presentan valores atípicos y en otros casos, se presentan en diferentes formatos. Para resolver estos problemas, se utilizaron técnicas de Minería de Datos que permitieron llevar a cabo el pre-procesamiento de los datos.

Debido a la amplia colección de instancias de prueba se limitó la selección de instancias y se realizó un muestreo o selección a priori bajo las siguientes restricciones:

a) De las 3 mejoras seleccionadas se obtuvieron 22 instancias de prueba, de las cuales 3 corresponden a las instancias resueltas en [58], 8 a las instancias resueltas en [19] y 11 corresponden a las instancias resueltas en [61].

b) Esta selección se realizó con el objetivo de replicar los experimentos de cada mejora y observar su comportamiento respecto al algoritmo K-means estándar y respecto a las otras mejoras, de manera que se puedan mostrar los casos en que cada mejora presenta resultados más ventajosos.

c) Otro aspecto importante es mostrar el comportamiento de los algoritmos al solucionar instancias en diferentes escenarios, para esto se realizó un diseño experimental descrito en la Sección 4.3.

En la Tabla 4.2 se describen las instancias reales utilizadas en el análisis comparativo, donde n es el número de registros, d representa el número de dimensiones y k indica el número de grupos experimentados.

Tabla 4.2 Instancias reales para el análisis comparativo

ID	Instancia	n	d	k
R1	Abalone	4177	7	200, 400, 600, 800, 1000
R2	Concrete C.	1030	8	100
R3	Iris	150	3	3, 5, 10, 20, 30, 40, 50
R4	Letter R.	20000	16	40, 50, 60, 70, 80, 90, 100
R5	New York	657308	2	400
R6	Skin S.	245057	3	100
R7	uci-sc	600	60	6
R8	Wind	6574	15	20, 40, 60, 80, 100, 120, 140, 160

Los datos sintéticos utilizados en la experimentación fueron generados a partir de las configuraciones indicadas en la Tabla 4.3, tomando como semilla la función de tiempo del sistema.

Tabla 4.3 Instancias sintéticas para el análisis comparativo

ID	Instancia	n	d	k	Rango de valores	Tipo de datos
S1	2,500	2500	2	50, 100, 200, 400, 800	1-50	Enteros
S2	10,000	10000	2	50, 100, 200, 400, 800	1-100	Enteros
S3	20,000	20000	2	50, 100, 200, 400, 800	1-200	Enteros
S4	40,000	40000	2	50, 100, 200, 400, 800	1-200	Enteros
S5	DSH1	6000	2	50	0-1	Enteros
S6	DSH2	6000	10	50	0-1	Reales
S7	DSH3	6000	25	50	0-1	Reales
S8	DSH4	6000	50	50	0-1	Reales
S9	DSH5	6000	100	50	0-1	Reales
S10	DSH6	6000	250	50	0-1	Reales
S11	DSH7	6000	500	50	0-1	Reales
S12	DSH8	6000	1000	50	0-1	Reales
S13	DSL1	60000	2	50	0-1	Reales
S14	DSL2	600000	2	50	0-1	Reales

4.5 Implementación y experimentación

4.5.1 Configuraciones de implementación

Se implementaron computacionalmente cuatro algoritmos, a saber, el algoritmo K-means estándar y las tres mejoras más relevantes seleccionadas en la Sección 4.1, éstas, fueron codificados en lenguaje de programación C.

Con el objetivo de estandarizar las implementaciones, se tomó como referencia el algoritmo K-means estándar [29] y se mantuvieron fijos los criterios de inicialización, cálculos de centroides y convergencia. La fase de inicialización consiste en la lectura de centroides iniciales generados para la ejecución de cada prueba, mientras que el criterio de convergencia se fijó en detener el algoritmo hasta que los centroides ya no cambien.

4.5.2 Configuraciones del equipo

Para la implementación y experimentación se utilizó un equipo Mac Mini con la siguiente configuración: Procesador Intel Core i7, 16 Gb en memoria RAM, sistema operativo Yosemite y compilador GCC versión 4.2.4.

4.5.3 Configuraciones de experimentación

Con base en el diseño experimental, se obtuvieron 61 configuraciones, es decir, 61 pruebas que fueron ejecutadas 30 veces por 4 algoritmos. El total de ejecuciones realizadas son 7,320 ejecuciones.

Las 30 ejecuciones se realizan con el objetivo de minimizar los errores debido a la aleatoriedad. Es importante mencionar que para cada una de las 30 ejecuciones por cada una de las 61 pruebas, se proporcionó el mismo conjunto de centroides iniciales a cada algoritmo, esto, con el fin de analizar sus resultados en igualdad de condiciones y reducir de esta manera las variables para la comparación.

4.5.4 Configuraciones de inicialización

Se utilizó un generador de centroides aleatorios para el cual, dada una instancia *d*-dimensional $D = \{x_1, x_2, \dots, x_n\}$ y un valor *k,* se obtuvieron *k* números aleatorios del conjunto de datos *D* y se mapearon sus valores a un archivo de texto plano.

Se generaron 1830 archivos de centroides iniciales, los cuales corresponden a 30 ejecuciones para cada una de las 61 pruebas realizadas en la experimentación.

4.5.5 Creación de herramientas

Para la creación de datos sintéticos se diseñó una herramienta que consiste en obtener como valores de entrada el número de objetos *n*, el número de dimensiones *d*, el número de grupos *k*, el rango de valores y tipo de datos (enteros o reales). Por otra parte, debido al gran número de ejecuciones en la experimentación, se diseñó una herramienta para la ejecución de las pruebas en serie. Esta herramienta recibe como entrada los archivos de centroides iniciales de los algoritmos en cada corrida y consiste en la ejecución de un Shell, facilitando el manejo de archivos y la escritura de los resultados.

4.5.6 Resultados experimentales

Debido al gran número de datos obtenidos por múltiples ejecuciones de los algoritmos, se realizó un post-procesamiento de los datos. Primeramente se concentraron los resultados de cada una de las 30 ejecuciones para cada prueba y posteriormente se promediaron dichos resultados, por lo tanto, se obtuvieron 244 resultados que corresponden a promedios de 30 ejecuciones de 61 pruebas realizadas por 4 algoritmos.

Los índices comparativos fueron calculados para cada uno de los resultados y se realizaron gráficas que apoyaron el proceso de análisis. Es importante señalar que previo a la fase de análisis experimental se realizó la validación estadística de nuestros resultados. En la Sección 4.7 se muestran los resultados por cada experimento y se

mencionan las observaciones más relevantes. Para más detalle en el Anexo B se muestran las tablas generales de los resultados.

4.6 Validación estadística

Dado que las poblaciones de interés son: 1) un conjunto de instancias de pruebas aplicables al algoritmo K-means y 2) un conjunto de resultados de SSE y tiempo al resolver un conjunto de instancias de prueba con diferentes algoritmos; se obtuvieron 4 muestras de 61 resultados en términos de SSE y tiempo de ejecución al resolver 22 instancias de prueba en diferentes escenarios con el algoritmo K-means estándar, *Early Classification*, *Enhanced K-means* y *Pattern Reduction*, respectivamente.

Con base en el marco estadístico propuesto, primero se evaluó el supuesto de normalidad de datos, para esto, se realizaron histogramas de frecuencia y se aplicó la prueba de *Kolmogorov-Smirnov* para cada muestra obtenida. Al analizar los resultados de dichas pruebas se determinó que con un nivel de confianza del 95% las muestras no presentan normalidad de datos ($p<0.05$).

En cuanto al supuesto de homocedasticidad de varianzas, se analizaron las muestras mediante la prueba de *Levene* y se obtuvo que las muestras respecto a soluciones de error al cuadrado obtenidas por la ejecución de 4 algoritmos presentan homogeneidad de varianzas ($p=0.986$), lo cual, confirma los resultados experimentales, ya que se observó que las mejoras presentan una pérdida mínima de calidad respecto a la versión estándar del algoritmo K-means. Por otra parte, se obtuvo que las muestras respecto a las soluciones de tiempo de ejecución obtenidas por 4 algoritmos no presentan homogeneidad de varianzas ($p<0.05$).

Al no cumplirse los supuestos paramétricos, nos enfocamos a la aplicación de pruebas no paramétricas, y con este enfoque se evaluaron los resultados de las 30 ejecuciones para cada prueba. Se aplicó la prueba de Wilcoxon estableciendo un nivel de confianza del 95% y las hipótesis:

H_0= Las diferencias en las soluciones de error al cuadrado obtenidas por la ejecución de dos algoritmos son debido a la aleatoriedad.

H_1= Las diferencias en las soluciones de error al cuadrado obtenidas por la ejecución de dos algoritmos no se deben a la aleatoriedad.

Los resultados obtenidos al realizar 366 pruebas de Wilcoxon sobre las muestras indican que con un nivel de confianza del 95%, el 99.2% de los resultados presentan un nivel de significancia $p<0.05$, por lo tanto, nuestros resultados experimentales no se deben a la aleatoriedad, sino a los enfoques de solución de cada algoritmo.

La validación de los resultados experimentales nos permitió determinar si existe diferencia estadísticamente significativa entre los algoritmos, para esto, se realizó la prueba de *Friedman*, mediante la cual se determinó que con un nivel de confianza del 95% los algoritmos mantienen una diferencia estadísticamente significativa ($p<0.05$). Como paso final, se analizaron las pruebas de Wilcoxon con el objetivo de determinar entre que pares de muestras se encontró dicha diferencia.

Un ordenamiento ascendente de la sumatoria del error al cuadrado y el tiempo de ejecución se realizó y se presenta en la Sección 4.7. Los resultados de las pruebas estadísticas realizadas se presentan de manera más detallada en el Anexo C.

4.7 Presentación de resultados y validación experimental

En las subsecciones siguientes se presentan los resultados para cada tipo de experimento y se mencionan los patrones de comportamiento más relevantes.

4.7.1 Resultados del experimento A

Mediante el análisis de los resultados se tuvieron observaciones importantes de modo que en la Figura 4.4 se muestra el comportamiento de los algoritmos al resolver la instancia R4 (experimento A), en la cual se aprecia una pérdida no significativa de la calidad respecto a la versión estándar del algoritmo K-means, mientras que, en la Figura 4.5 se observa que las variantes implementadas presentan una reducción de tiempo de ejecución importante respecto al algoritmo K-means estándar. De manera general se observa que para la mayoría de los casos el algoritmo *Pattern Reduction* presenta mejores resultados en tiempo de ejecución, sin embargo, su pérdida de calidad es mayor que en *Early Classification* y *Enhanced K-means.*

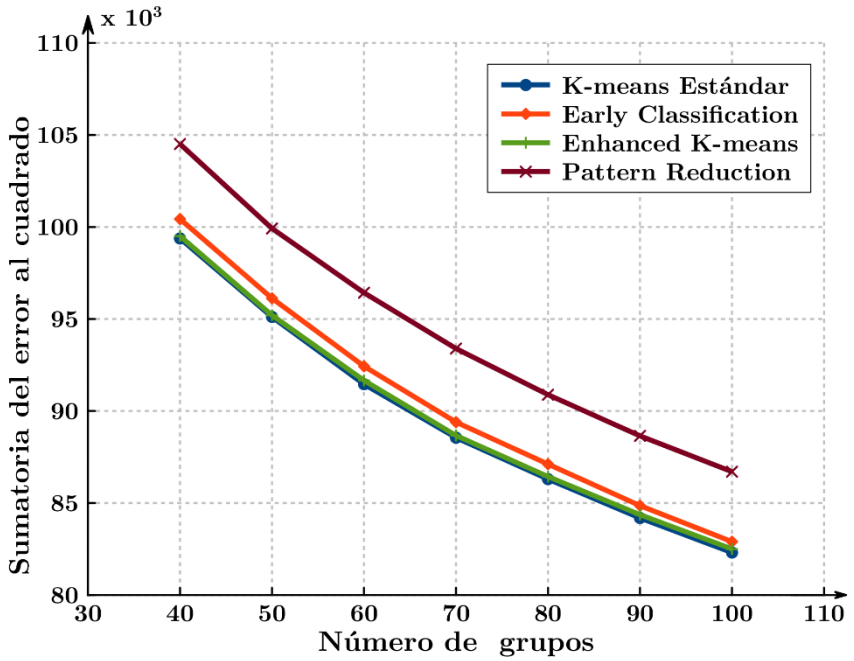


Figura 4.4 Comportamiento de la SSE de los algoritmos implementados al resolver la instancia *Letters* con incrementos en el número de grupos

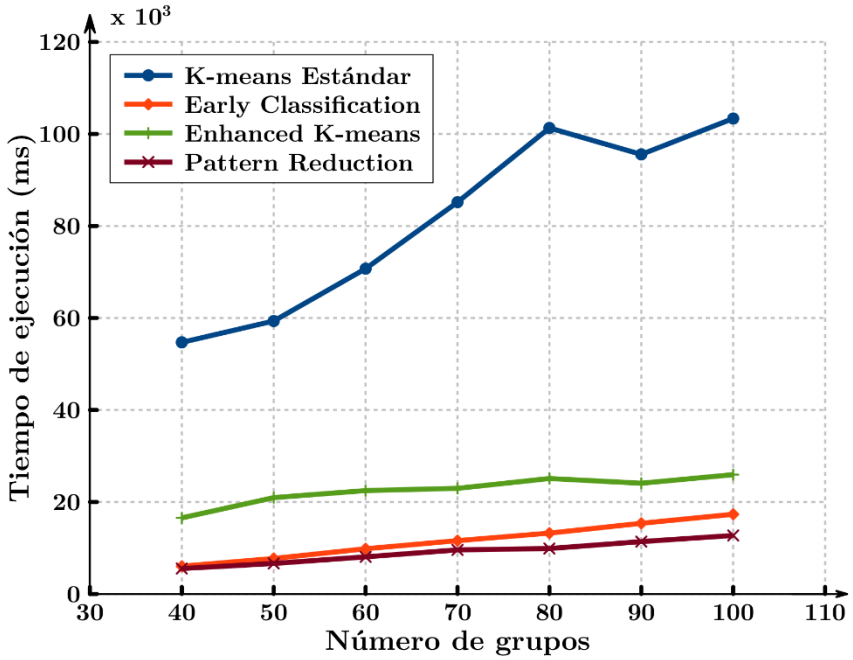


Figura 4.5 Comportamiento del tiempo de ejecución de los algoritmos implementados al resolver la instancia *Letters* con incrementos en el número de grupos

En la Tabla 4.4 se muestra la pérdida de calidad de cada variante en comparación con la versión estándar del algoritmo K-means. En este experimento se observa que a medida que se incrementan los grupos, la pérdida de calidad aumenta con *Enhanced K-means*, mientras que en *Early Classification* y *Pattern Reduction* la pérdida de calidad disminuye. Las cifras sombreadas, muestran los mejores resultados en términos de pérdida de calidad, donde las cifras más cercanas a cero expresan una similitud a la función objetivo del algoritmo estándar.

Tabla 4.4 Resultados del experimento A en términos de porcentaje de pérdida de calidad

Instancia	Grupos	K-means Estándar	Early Classification		Enhanced K-means		Pattern Reduction	
		SSE	SSE	% ε	SSE	% ε	SSE	% ε
Abalone	200	181	181	-0.27	182	-0.79	190	-4.92
Abalone	400	152	152	-0.08	153	-1.03	157	-3.70
Abalone	600	135	135	-0.01	136	-1.03	139	-3.14
Abalone	800	122	122	0.00	123	-0.98	126	-2.72
Abalone	1000	112	112	0.00	113	-0.89	115	-2.54
Letter R.	40	99377	100434	-1.06	99504	-0.13	104510	-5.16
Letter R.	50	95111	96123	-1.06	95201	-0.09	99920	-5.06
Letter R.	60	91456	92430	-1.07	91659	-0.22	96426	-5.43
Letter R.	70	88550	89394	-0.95	88667	-0.13	93393	-5.47
Letter R.	80	86296	87111	-0.94	86435	-0.16	90886	-5.32
Letter R.	90	84185	84856	-0.80	84362	-0.21	88646	-5.30
Letter R.	100	82305	82897	-0.72	82499	-0.24	86701	-5.34
Wind	20	68738	69135	-0.58	68758	-0.03	70693	-2.84
Wind	40	62706	63059	-0.56	62790	-0.13	64664	-3.12
Wind	60	59685	59917	-0.39	59812	-0.21	61496	-3.03
Wind	80	57611	57771	-0.28	57747	-0.24	59186	-2.73
Wind	100	55996	56104	-0.19	56128	-0.24	57496	-2.68
Wind	120	54683	54763	-0.15	54825	-0.26	56118	-2.62
Wind	140	53594	53649	-0.10	53758	-0.31	54981	-2.59
Wind	160	52609	52659	-0.10	52814	-0.39	54015	-2.67

En la Tabla 4.5 se muestra el porcentaje de reducción de tiempo por cada algoritmo, de la misma forma que en la calidad, se observa que a medida que se incrementan los grupos el porcentaje de reducción de tiempo en *Early Classification* y *Pattern Reduction* disminuye, mientras que, en *Enhanced K-means* se obtiene una ganancia de tiempo mayor cuando se aumenta el número de grupos en las instancias Letter R. y Wind.

Tabla 4.5 Resultados del experimento A en términos de porcentaje de reducción de tiempo

Instancia	Grupos	K-means Estándar	Early Classification		Enhanced K-means		Pattern Reduction	
		Tiempo	Tiempo	% τ	Tiempo	% τ	Tiempo	% τ
Abalone	200	6273	3742	40.34	1487	76.29	1243	80.18
Abalone	400	8598	6470	24.75	2084	75.76	2233	74.02
Abalone	600	9871	8299	15.92	2556	74.10	3314	66.42
Abalone	800	10869	9566	11.99	3066	71.79	3314	69.50
Abalone	1000	12667	11096	12.41	3514	72.26	5030	60.29
Letter R.	40	54701	6035	88.97	16544	69.75	5531	89.89
Letter R.	50	59343	7719	86.99	20950	64.70	6642	88.81
Letter R.	60	70722	9831.80	86.10	22481	68.21	8073	88.58
Letter R.	70	85177	11598.6	86.38	22966	73.04	9588	88.74
Letter R.	80	101305	13233.4	86.94	25106	75.22	9885	90.24
Letter R.	90	95564	15368.2	83.92	24077	74.81	11388	88.08
Letter R.	100	103365	17330.4	83.23	2594	74.90	12725	87.69
Wind	20	6005	770.900	87.16	2367	60.58	647	89.22
Wind	40	11422	1884.20	83.50	3382	70.39	1305	88.57
Wind	60	15584	3268.50	79.03	3888	75.05	1778	88.59
Wind	80	16713	4671.80	72.05	4254	74.54	2347	85.95
Wind	100	18612	6092.90	67.26	4785	74.29	3047	83.63
Wind	120	19494	7675.80	60.63	5124	73.71	3483	82.13
Wind	140	21278	8988.80	57.76	5328	74.96	3907	81.64
Wind	160	24923	10964.1	56.01	5708	77.10	4346	82.56

Por otra parte, se observó que a medida que el número de grupo se aproxima al número de objetos (como en la instancia Abalone con k=800 y k=1000), el algoritmo *Early Classification* presenta una pérdida de calidad aproximada al 0% (ver Tabla 4.4), sin embargo, el porcentaje de reducción de tiempo es menor al 15% (ver Tabla 4.5).

El comportamiento descrito anteriormente se puede observar en las Figuras 4.6 y 4.7 al resolver la instancia S1 con n=2500, d=2 e incrementos en el número de grupos k=50, k=100, k=200, k=400, k=800. Se estimó la distribución de datos por grupo y se obtuvo que para cada caso con este comportamiento, la proporción de objetos por grupo es igual a 6, por lo tanto se recomienda tomar en cuenta que dicha proporción sea mayor a 6 para obtener mejores resultados en cuanto al tiempo de solución.

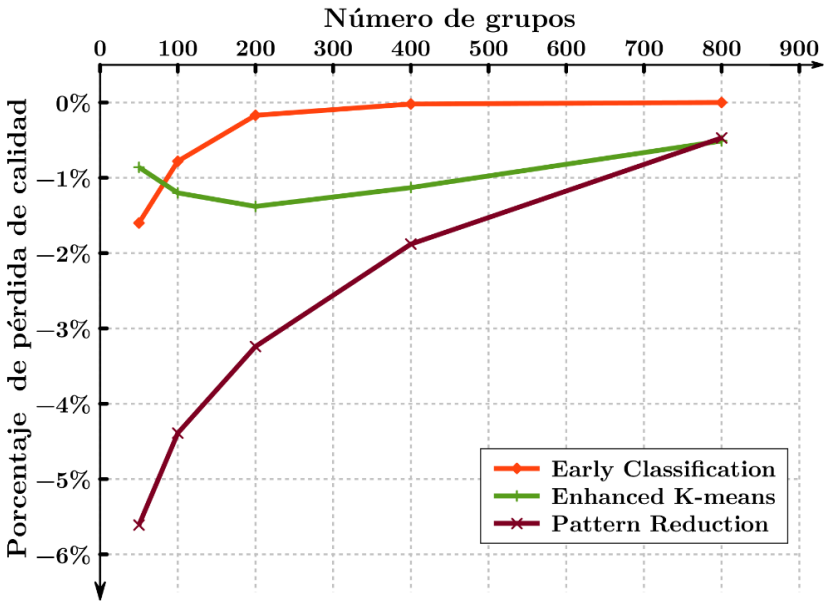


Figura 4.6 Comportamiento de los algoritmos respecto al porcentaje de pérdida de calidad al resolver la instancia S1

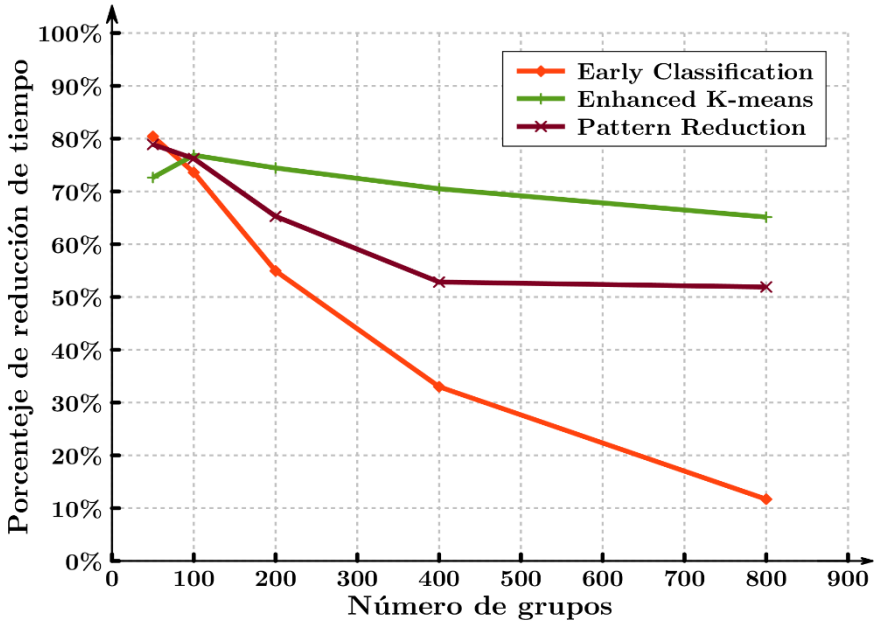


Figura 4.7 Comportamiento de los algoritmos respecto al porcentaje de reducción de tiempo al resolver la instancia S1

4.7.2 Resultados del experimento B

Respecto al comportamiento de los algoritmos con el tipo de experimento B, se observa que a medida que se incrementa el número de objetos manteniendo una k=100, los algoritmos presentan una reducción considerable en tiempo de ejecución, de los cuales, para los casos de prueba, la mejora *Early Classification* obtuvo una reducción de tiempo mayor sin pérdida significativa de la calidad, esto, debido a su enfoque de exclusión de objetos de futuros cálculos (ver Figura 4.8 y Figura 4.9).

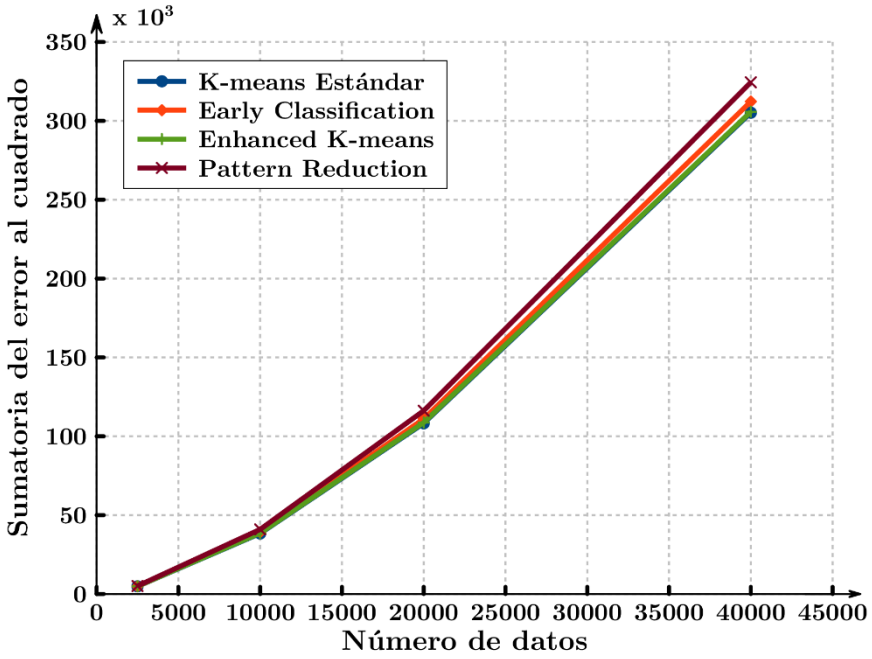


Figura 4.8 Comportamiento de los algoritmos en términos de SSE al resolver las instancias S1, S2, S3 y S4 con k=100

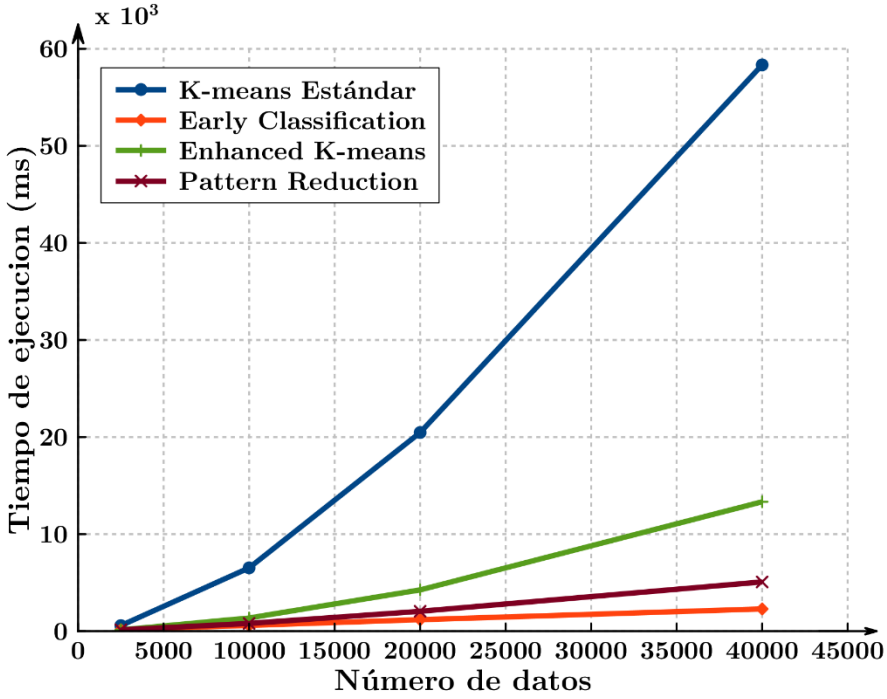


Figura 4.9 Comportamiento de los algoritmos en términos de tiempo de ejecución al resolver las instancias S1, S2, S3 y S4 con k=100

La Tabla 4.6 representa la pérdida de calidad de cada variante en comparación con la versión estándar del algoritmo K-means con la ejecución del experimento B. En este experimento se observa que el algoritmo *Enhanced K-means* presenta una pérdida de calidad menor a las otras variantes, sin embargo, el algoritmo *Early Classification* muestra un buen desempeño con una pérdida de calidad menor al 3%.

Tabla 4.6 Resultados del experimento B en términos de porcentaje de pérdida de calidad

Instancia	Grupos	K-means Estándar	Early Classification		Enhanced K-means		Pattern Reduction	
		SSE	SSE	% ε	SSE	% ε	SSE	% ε
2500	50	6806	6915	-1.60	6865	-0.86	7188	-5.61
2500	100	4852	4890	-0.78	4910	-1.20	5065	-4.39
2500	200	3451	3457	-0.17	3499	-1.38	3563	-3.24
2500	400	2444	2445	-0.02	2472	-1.13	2490	-1.88
2500	800	1704	1704	0.00	1713	-0.51	1713	-0.52
10000	50	54177	55614	-2.65	54355	-0.33	58240	-7.50
10000	100	38363	39214	-2.22	38625	-0.68	40978	-6.82
10000	200	27255	27624	-1.35	27485	-0.84	28756	-5.51
10000	400	19367	19484	-0.60	19587	-1.13	20246	-4.54
10000	800	13784	13796	-0.09	13958	-1.26	14216	-3.13
20000	50	153163	157823	-3.04	153591	-0.28	165343	-7.95
20000	100	108233	111046	-2.60	108719	-0.45	116204	-7.37
20000	200	76673	78142	-1.92	77105	-0.56	81451	-6.23
20000	400	54414	55031	-1.13	54840	-0.78	57293	-5.29
20000	800	38690	38866	-0.45	39108	-1.08	40323	-4.22
40000	50	432573	444106.1	-2.67	433020	-0.10	461749	-6.74
40000	100	305172	312266	-2.32	305787	-0.20	324437	-6.31
40000	200	216011	220729	-2.18	216771	-0.35	230194	-6.57
40000	400	153156	155575	-1.58	153905	-0.49	161927	-5.73
40000	800	108735	109837	-1.01	109545	-0.74	114426	-5.23

Respecto al porcentaje de reducción de tiempo en la Tabla 4.7 se puede observar una superioridad del algoritmo *Early Classification*. Cómo se mencionó en el experimento A, se puede contrastar que cuando n es menor y el número de grupos se eleva, el algoritmo aumenta su tiempo de ejecución como en los casos de n=2500 con k= 100, 200, 400 y 800 y n=10000 y 20000 objetos con k=100. Por el contrario, el algoritmo *Enhanced K-means* tiene un buen desempeño al incrementarse el número de grupos y menor número de objetos.

Tabla 4.7 Resultados del experimento B en términos de porcentaje de reducción de tiempo

Instancia	Grupos	K-means Estándar	Early Classification		Enhanced K-means		Pattern Reduction	
		Tiempo	Tiempo	% τ	Tiempo	% τ	Tiempo	% τ
2500	50	368	72	80.42	100	72.63	77	78.89
2500	100	578	152	73.59	133	76.85	137	76.22
2500	200	716	322	54.94	183	74.44	248.	65.28
2500	400	930	623	33.00	274	70.50	438	52.83
2500	800	1273	1124	11.70	443	65.13	612	51.90
10000	50	3958	284	92.80	1009	74.49	446	88.74
10000	100	6523	595	90.87	1361	79.13	802	87.70
10000	200	8372	1301	84.45	1678	79.95	1426	82.96
10000	400	11269	2888	74.37	2189	80.58	2404	78.66
10000	800	13430	6244	53.51	2971	77.88	4203	68.70
20000	50	10552	559	94.69	2782	73.63	1084	89.72
20000	100	20464	1188	94.19	4254	79.21	2049	89.99
20000	200	28725	2533	91.18	5320	81.48	3578	87.54
20000	400	40085	5599	86.03	6700	83.28	6056	84.89
20000	800	48673	12575	74.16	8813	81.89	10775	77.86
40000	50	25790	1116	95.67	8296	67.83	2716	89.47
40000	100	58343	2293	96.07	13334	77.15	5076	91.30
40000	200	96967	5146	94.69	17368	82.09	9087	90.63
40000	400	131262	10965	91.65	21792.	83.40	15607	88.11
40000	800	170438	24507	85.62	27710	83.74	27003	84.15

4.7.3 Resultados del experimento C

El experimento C consistió en probar el desempeño de los algoritmos y observar su comportamiento al incrementar el número de dimensiones desde 2 a 1000 para el mismo n=6000 y k=50.

La Tabla 4.8 muestra que a medida que se incrementa el número de dimensiones, se disminuye la pérdida de calidad en las tres variantes implementadas.

Tabla 4.8 Resultados del experimento C en términos de porcentaje de pérdida de calidad

Instancia	*n*	*d*	*k*	K-means Estándar	Early Classification		Enhanced K-means		Pattern Reduction	
				SSE	SSE	% ε	SSE	% ε	SSE	% ε
DSH1	6000	2	50	320	330	-2.99	323	-1.01	343	-7.30
DSH2	6000	10	50	3781	3803	-0.57	3787	-0.14	3879	-2.59
DSH3	6000	25	50	7503	7508	-0.07	7510	-0.09	7590	-1.16
DSH4	6000	50	50	11345	11345	0.00	11350	-0.04	11426	-0.71
DSH5	6000	100	50	224	229	-2.10	227	-1.18	239	-6.84
DSH6	6000	250	50	26828	26828	0.00	26832	-0.01	26875	-0.17
DSH7	6000	500	50	38261	38261	0.00	38263	-0.01	38289	-0.07
DSH8	6000	1000	50	54312	54312	0.00	54314	0.00	54324	-0.02

Por otra parte, en la Tabla 4.9 se puede observar que a pesar de presentar menor pérdida de calidad, *Early Classification* y *Enhanced K-means* aumentan su tiempo de ejecución al incrementarse el número de dimensiones; mientras que el algoritmo *Pattern Reduction* presenta reducciones de tiempo superiores al 70% y una pérdida de calidad menor al 2% para los casos DSH3, DSH4, DSH6, DSH7 y DSH8.

Tabla 4.9 Resultados del experimento C en términos de porcentaje de reducción de tiempo

Instancia	*n*	*d*	*k*	K-means Estándar	Early Classification		Enhanced K-means		Pattern Reduction	
				Tiempo	Tiempo	% τ	Tiempo	% τ	Tiempo	% τ
DSH1	6000	2	50	1579	172	89.1	404	74.4	251	84.1
DSH2	6000	10	50	9142	1628	82.2	2598	71.5	1262	86.2
DSH3	6000	25	50	18264	6748	63.0	5667	68.9	3085	83.1
DSH4	6000	50	50	27555	16899	38.7	9291	66.2	5678	79.4
DSH5	6000	100	50	2556	374	85.4	519	79.6	436	82.9
DSH6	6000	250	50	81986	75572	7.82	29632	63.8	22475	72.6
DSH7	6000	500	50	124854	120560	3.44	49131	60.6	42308	66.1
DSH8	6000	1000	50	206349	195695	5.16	81387	60.5	75589	63.4

4.7.4 Resultados del análisis de la instancia Iris

La instancia Iris (R3), es un conjunto de datos comúnmente reconocido por la comunidad científica, la cual, consta de 150 objetos en 3 dimensiones que expresan medidas de la longitud del sépalo, ancho del sépalo y longitud del pétalo. Es importante mencionar que esta instancia se ha estudiado en muchos trabajos de agrupamiento de datos, donde su agrupamiento óptimo es conocido e indica las especies de Iris (ver Figura 4.10).

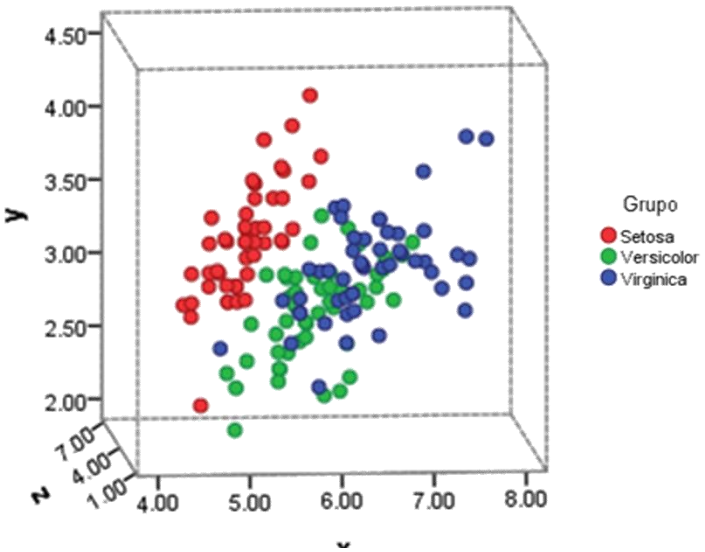
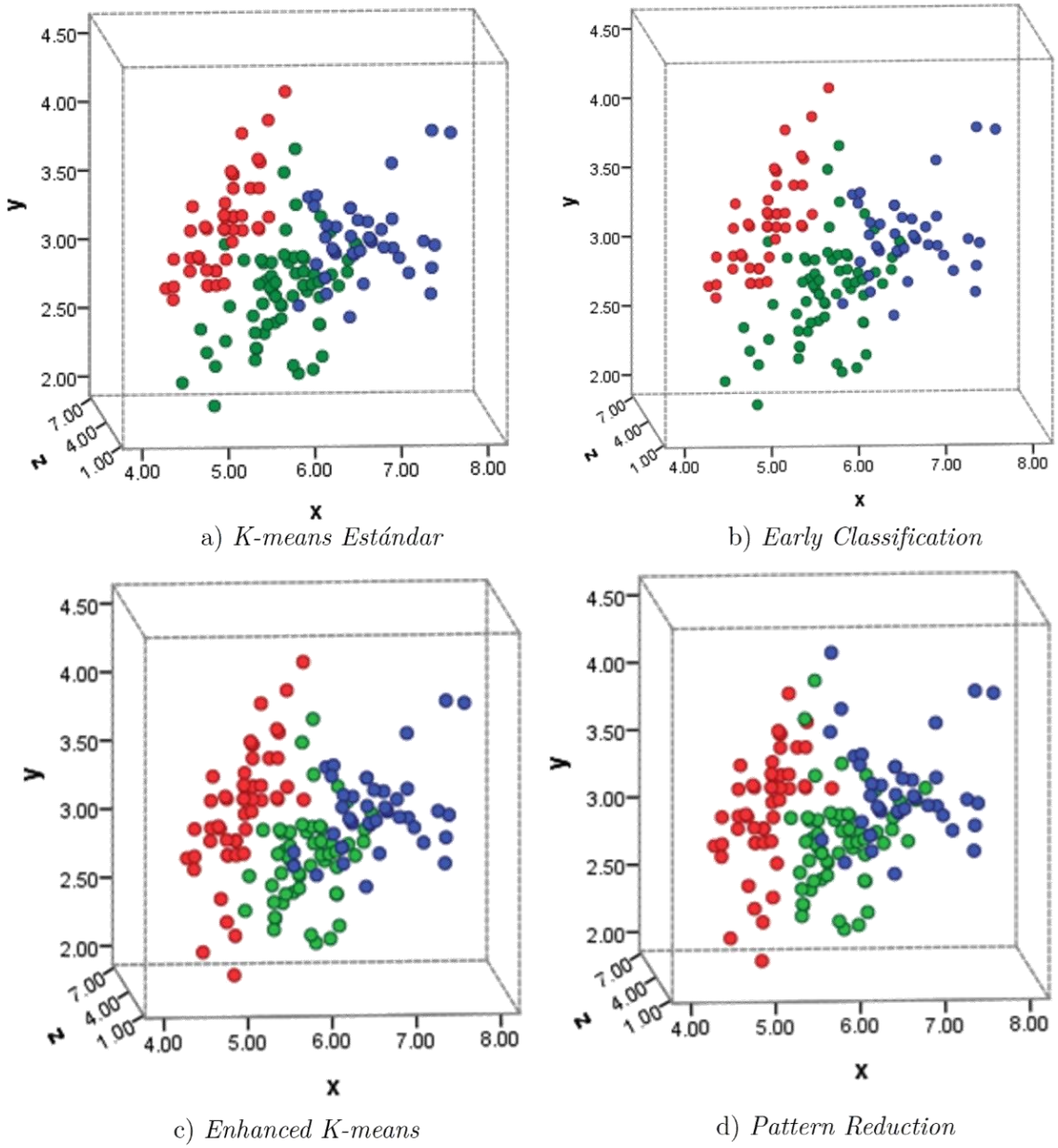


Figura 4.10 Distribución de datos de la instancia Iris

Con la finalidad de mostrar el comportamiento del algoritmo K-means estándar y las mejoras seleccionadas en la Sección 4.1, se realizó una ejecución de cada algoritmo para resolver la instancia Iris con la siguiente configuración: n=150, d=3, k=3, y los centroides iniciales m_1= (7.7, 2.6, 6.9), m_2= (5.2, 4.1, 1.5) y m_3= (5.6, 3.0, 4.5).

En la Figura 4.11 se muestra el agrupamiento final al resolver la instancia con cada algoritmo. Es relevante mencionar que de acuerdo a la solución óptima de la instancia, el algoritmo K-means estándar agrupó 134 objetos correctamente, mientras que *Early Classification*, *Enhanced K-means* y *Pattern Reduction* agruparon correctamente 129, 125 y 116 objetos, respectivamente.



a) *K-means Estándar*

b) *Early Classification*

c) *Enhanced K-means*

d) *Pattern Reduction*

Figura 4.11 Solución de la instancia Iris por cada algoritmo

Se realizaron otras ejecuciones para observar más a detalle el comportamiento de las mejoras respecto a la versión estándar del algoritmo K-means. Los resultados de promedios de 30 ejecuciones para cada una de las pruebas en términos de porcentaje de

pérdida de calidad y porcentaje de reducción de tiempo se muestran en la Tabla 4.10. Las celdas marcadas con amarillo indican los mejores resultados en términos de porcentaje de pérdida de calidad, mientas que las celdas marcadas con rojo, indican los mejores resultados en términos de porcentaje de reducción de tiempo.

Se puede observar una ventaja significativa de la mejora *Early Classification* al obtener en la mayoría de los casos una menor pérdida de calidad, sin embargo, su reducción de tiempo es menor respecto a las otras mejoras. Por otra parte se puede observar un buen comportamiento de la mejora *Enhanced K-means,* donde, su pérdida de calidad no es significativa y presenta un tiempo de reducción de hasta 57.98%. Finalmente, es destacable el tiempo de reducción de la mejora *Pattern Reduction*, sin embargo, como se ha venido analizando, ésta, presenta una pérdida de calidad elevada, donde, su pérdida de calidad mínima es aproximadamente dos veces la pérdida de calidad máxima de *Early Classification* o *Enhanced K-means*.

Tabla 4.10 Resultados de las pruebas con la instancia R3 (Iris)

Instancia	**Grupos**	**Early Classification**		**Enhanced K-means**		**Pattern Reduction**	
		% ε	% T	% ε	% T	% ε	% T
R3 (Iris)	3	-0.25	33.91	-0.10	19.23	-5.72	52.47
	5	-1.11	49.47	-0.05	35.69	-6.13	64.74
	10	-0.51	42.62	-1.09	47.02	-5.12	59.58
	20	-0.08	34.58	-0.97	54.47	-3.37	55.70
	30	-0.05	27.80	-1.29	57.14	-4.35	54.41
	40	-0.11	22.73	-1.20	57.98	-4.21	52.38
	50	0.00	13.21	-0.85	53.72	-2.65	40.37

4.7.5 Resultados de pruebas con instancias S13 y S14

De acuerdo con las pruebas realizadas en el experimento B, se determinó que la mejora *Early Classification* presenta una ventaja al trabajar con instancias con gran cantidad de objetos. Para continuar con este análisis en la Tabla 4.11 se presentan los resultados de promedios de 30 ejecuciones al resolver las instancias S13 y S14 con valores de tipo real con una distribución normal en un rango de valores de 0-1.

Como se puede observar, en términos de porcentaje de reducción de tiempo es sobresaliente la ventaja que presenta la mejora *Early Classification,* ya que llegó a reducir más del 97% del costo computacional con una pérdida de calidad de tan sólo el 3.11%. En amarillo se marcan los mejores resultados respecto al porcentaje de pérdida de calidad, donde, *Enhanced K-means* presenta un buen desempeño al reducir más del 50 % del tiempo de solución.

A pesar de obtener una reducción de tiempo por encima del 90%, *Pattern Reduction* mantiene elevada su pérdida de calidad, sin embargo, en la mayoría de los casos donde se resuelven instancias con una distribución normal, ésta, no eleva su pérdida de calidad a más del 10%. Otros casos con diferente distribución de datos son analizados en la siguiente subsección.

Tabla 4.11 Resultados de las pruebas con las instancias S13 y S14

Instancia	n	d	k	Early Classification		Enhanced K-means		Pattern Reduction	
				% ε	% τ	% ε	% τ	% ε	% τ
S13 (DSL1)	60000	2	50	-3.11	96.1	-0.08	66.7	-8.05	90.2
S14 (DSL2)	600000	2	50	-3.07	97.8	-0.02	53.9	-7.43	90.0

4.7.6 Resultados de pruebas con instancias con mayor dispersión de datos

Es destacable mencionar que se realizaron pruebas con instancias que presentan datos con mayor dispersión como el caso de R2, R5, R6 y R7, donde el algoritmo *Pattern Reduction* obtiene mejores resultados de tiempo (ver Tabla 4.13), sin embargo, en términos de porcentaje de pérdida de calidad presenta una pérdida significativa de hasta un 24.9% (ver Tabla 4.12), esto, debido a su enfoque de compresión de objetos que se encuentran cercanos a la media. Cuando los datos no cumplen una normalidad, se excluyen objetos de futuros cálculos erróneamente, por lo tanto, este algoritmo tiene mejores resultados con instancias que presentan datos normales.

Respecto a los resultados observados en estas pruebas, se determina que el algoritmo *Enhanced K-means* presenta mejores resultados al reducir hasta 78% de tiempo con una pérdida de calidad menor al 2%, esto, debido a que la heurística de optimización se enfoca en la reducción del número de grupos.

Tabla 4.12 Resultados de porcentaje de pérdida de calidad al resolver instancias con mayor dispersión de datos

Instancia	Grupos	K-means Estándar	Early Classification		Enhanced K-means		Pattern Reduction	
		SSE	SSE	% ε	SSE	% ε	SSE	% ε
Concrete	100	42890	42941	-0.12	43719	-1.93	45062	-5.06
Skin	100	1881762	1965730	-4.46	1898027	-0.86	2351598	-24.9
New York	100	1439	1501	-4.32	1460	-1.44	1781	-23.8
Uci-Sc	6	23420	23431	-0.05	23434	-0.06	23930	-2.18

Tabla 4.13 Resultados de porcentaje de reducción de tiempo al resolver instancias con mayor dispersión de datos

Instancia	Grupos	K-means Estándar	Early Classification		Enhanced K-means		Pattern Reduction	
		Tiempo	Tiempo	% τ	Tiempo	% τ	Tiempo	% τ
Concrete	100	345	243	29.62	111	67.65	135	60.76
Skin	100	401905	60332	84.99	99854	75.15	23224	94.22
New York	100	2015454	539107	73.25	423637	78.98	213850	89.39
Uci-Sc	6	80	42	47.22	51	35.72	38	52.50

4.7.7 Casos de dominancia de las mejoras del algoritmo K-means

Derivado de las observaciones, se determinó que los experimentos realizados eran suficientes para soportar el análisis comparativo, por lo tanto, en la Tabla 4.14 y la Tabla 4.15 se presenta un ordenamiento ascendente de la sumatoria del error al cuadrado por cada mejora en la solución de cada prueba. Este ordenamiento expresa los casos de dominancia de cada mejora. En la columna 1 se mencionan las instancias de prueba. En las columnas 2, 3 y 4 se describen las configuraciones de cada prueba mediante los valores n, d, y k, los cuales expresan el número de objetos, número de dimensiones y números de grupos, respectivamente. En la columna 5 se muestra un ordenamiento donde el algoritmo asociado a la izquierda presenta mejores resultados, es decir menor pérdida de calidad.

De forma general se puede observar que los algoritmos *Enhanced K-means* y *Early Classification* predominan, obteniendo mayor número de casos de dominancia: 33 y 28 casos de dominancia, respectivamente.

De los 33 casos de dominancia de la mejora *Enhanced K-means*, 15 casos corresponden a la solución de instancias reales (R3, R4 y R8) y 18 casos a la solución de instancias sintéticas (S2, S3, S4, S13 y S14). Mediante el análisis de estos casos se observó que dicha ventaja se debe a que las instancias se agruparon en grandes valores de k, donde $k<n$ y $k<d$.

De los 28 casos de dominancia de la mejora *Early Classification*, 16 casos corresponden a la solución de instancias reales (R1, R3, R5, R6, R7 y algunos casos con R8) y 12 casos a la solución de instancias sintéticas (S1, S2, S7, S8, S10, S11 y S12). Mediante el análisis de estos casos se observó que dicha ventaja se debe a que las instancias resueltas presentan grandes cantidades de objetos y se agrupan en valores de k proporcionales. Un patrón de comportamiento relevante se observó en algunos casos como en R1, R3 y S1, donde, se estimó el número de objetos por grupo y se identificó que cuando esta proporción es muy pequeña (en la mayoría de los casos menor a 10 objetos por grupo), la mejora *Early Classification* presenta un comportamiento similar al algoritmo estándar donde su pérdida de calidad fue aproximada a 0% y su reducción de tiempo fue menor al 30%.

Tabla 4.14 Ordenamiento ascendente por cada instancia real en función de la SSE (casos de dominancia)

Instancia	***n***	***d***	***k***	**Ordenamiento ascendente de la SSE (casos de dominancia)**
R1	4177	7	200	Early Classification < Enhanced K-means < Pattern Reduction
R1	4177	7	400	Early Classification < Enhanced K-means < Pattern Reduction
R1	4177	7	600	Early Classification < Enhanced K-means < Pattern Reduction
R1	4177	7	800	Early Classification < Enhanced K-means < Pattern Reduction
R1	4177	7	1000	Early Classification < Enhanced K-means < Pattern Reduction
R2	1030	8	100	Early Classification < Enhanced K-means < Pattern Reduction
R3	150	3	3	Enhanced K-means < Early Classification < Pattern Reduction
R3	150	3	5	Enhanced K-means < Early Classification < Pattern Reduction
R3	150	3	10	Early Classification < Enhanced K-means < Pattern Reduction
R3	150	3	20	Early Classification < Enhanced K-means < Pattern Reduction
R3	150	3	30	Early Classification < Enhanced K-means < Pattern Reduction
R3	150	3	40	Early Classification < Enhanced K-means < Pattern Reduction
R3	150	3	50	Early Classification < Enhanced K-means < Pattern Reduction
R4	20000	16	40	Enhanced K-means < Early Classification < Pattern Reduction
R4	20000	16	50	Enhanced K-means < Early Classification < Pattern Reduction
R4	20000	16	60	Enhanced K-means < Early Classification < Pattern Reduction
R4	20000	16	70	Enhanced K-means < Early Classification < Pattern Reduction
R4	20000	16	80	Enhanced K-means < Early Classification < Pattern Reduction
R4	20000	16	90	Enhanced K-means < Early Classification < Pattern Reduction
R4	20000	16	100	Enhanced K-means < Early Classification < Pattern Reduction
R5	657308	2	400	Enhanced K-means < Early Classification < Pattern Reduction
R6	245057	3	100	Enhanced K-means < Early Classification < Pattern Reduction
R7	600	60	6	Early Classification < Enhanced K-means < Pattern Reduction
R8	65740	15	20	Enhanced K-means < Early Classification < Pattern Reduction
R8	65740	15	40	Enhanced K-means < Early Classification < Pattern Reduction
R8	65740	15	60	Enhanced K-means < Early Classification < Pattern Reduction
R8	65740	15	80	Enhanced K-means < Early Classification < Pattern Reduction
R8	65740	15	100	Early Classification < Enhanced K-means < Pattern Reduction
R8	65740	15	120	Early Classification < Enhanced K-means < Pattern Reduction
R8	65740	15	140	Early Classification < Enhanced K-means < Pattern Reduction
R8	65740	15	160	Early Classification < Enhanced K-means < Pattern Reduction

Tabla 4.15 Ordenamiento ascendente por cada instancia sintética en función de la SSE (casos de dominancia)

.Instancia	n	d	k	Ordenamiento ascendente de la SSE (casos de dominancia)
S1	2500	2	50	Enhanced K-means < Early Classification < Pattern Reduction
S1	2500	2	100	Early Classification < Enhanced K-means < Pattern Reduction
S1	2500	2	200	Early Classification < Enhanced K-means < Pattern Reduction
S1	25000	2	400	Early Classification < Enhanced K-means < Pattern Reduction
S1	25000	2	800	Early Classification < Enhanced K-means < Pattern Reduction
S2	10000	2	50	Enhanced K-means < Early Classification < Pattern Reduction
S2	10000	2	100	Enhanced K-means < Early Classification < Pattern Reduction
S2	10000	2	200	Enhanced K-means < Early Classification < Pattern Reduction
S2	10000	2	400	Early Classification < Enhanced K-means < Pattern Reduction
S2	10000	2	800	Early Classification < Enhanced K-means < Pattern Reduction
S3	20000	2	50	Enhanced K-means < Early Classification < Pattern Reduction
S3	20000	2	100	Enhanced K-means < Early Classification < Pattern Reduction
S3	20000	2	200	Enhanced K-means < Early Classification < Pattern Reduction
S3	20000	2	400	Enhanced K-means < Early Classification < Pattern Reduction
S3	20000	2	800	Early Classification < Enhanced K-means < Pattern Reduction
S4	40000	2	50	Enhanced K-means < Early Classification < Pattern Reduction
S4	40000	2	100	Enhanced K-means < Early Classification < Pattern Reduction
S4	40000	2	200	Enhanced K-means < Early Classification < Pattern Reduction
S4	40000	2	400	Enhanced K-means < Early Classification < Pattern Reduction
S4	40000	2	800	Enhanced K-means < Early Classification < Pattern Reduction
S5	6000	2	50	Enhanced K-means < Early Classification < Pattern Reduction
S6	6000	10	50	Enhanced K-means < Early Classification < Pattern Reduction
S7	6000	25	50	Early Classification < Enhanced K-means < Pattern Reduction
S8	6000	50	50	Early Classification < Enhanced K-means < Pattern Reduction
S9	6000	100	50	Enhanced K-means < Early Classification < Pattern Reduction
S10	6000	250	50	Early Classification < Enhanced K-means < Pattern Reduction
S11	6000	500	50	Early Classification < Enhanced K-means < Pattern Reduction
S12	6000	1000	50	Early Classification < Enhanced K-means < Pattern Reduction
S13	60000	2	50	Enhanced K-means < Early Classification < Pattern Reduction
S14	600000	2	50	Enhanced K-means < Early Classification < Pattern Reduction

Al igual que en las tablas anteriores, en la Tabla 4.16 y la Tabla 4.17 se presenta un ordenamiento del desempeño de las mejoras en términos de tiempo de ejecución por cada instancia real y sintética, respectivamente. Las columnas de cada tabla presentan las mismas descripciones dadas anteriormente y se puede observar que para la mayoría de los casos el algoritmo *Pattern Reduction* presenta menores tiempos de ejecución, sin

embargo, como se analizó anteriormente, su pérdida de calidad es mayor respecto a las mejoras *Early Classification* y *Enhanced K-means*.

Los casos de dominancia observados son 29 casos por *Pattern Reduction*, 17 casos por *Early Classification* y 15 casos por *Enhanced K-means*.

De los 29 casos de dominancia por *Pattern Reduction*, 23 corresponden a la solución de instancias reales (R3, R4, R5, R6, R7 y R8) y 6 a la solución de instancias sintéticas (S6, S7, S8, S10, S11, S12). Mediante el análisis de estos casos se observó que de acuerdo al método de discriminación existen objetos que se discriminan tempranamente sobre todo si no presentan una distribución normal o los centroides iniciales no son tratados. Para los casos que presentan una distribución normal se obtuvieron pérdidas de calidad no mayores a 10% y tiempos de reducción por arriba del 60%, por lo tanto esta mejora se recomienda para los casos en que se quiera obtener una solución rápida, sobre todo en instancias con grandes dimensiones y distribución normal.

Los 17 casos de dominancia por *Early Classification* corresponden a la solución de las instancias sintéticas (S2, S3, S4, S5 y S9). Mediante su análisis se observó que esta mejora presenta muy buenos resultados, donde el porcentaje de reducción promedio oscila entre el 70% y 90% con pérdidas de calidad menores al 4%. Ésta se presenta como una alternativa en la solución de instancias con grandes cantidades de objetos con un costo computacional reducido y una pérdida de calidad no significativa.

Por último, de los 15 casos de dominancia por *Enhanced K-means,* 8 corresponden a la solución de instancias reales (R1, R2, R5, R6 y R7) y 7 corresponden a la solución de instancias sintéticas (S1 y S3). Mediante el análisis de estos casos se observó que éste algoritmo presenta un buen desempeño, ya que debido a su método de discriminación, sólo realiza dos cálculos de distancia y una comparación por cada objeto, por lo que no arriesga la calidad de manera significativa. Para la mayoría de los casos se observó una pérdida de calidad menor al 1.93 % y reducciones de tiempo de hasta el 83.4%. A pesar de que la reducción de tiempo no es la mejor, es destacable la calidad de la solución que esta mejora presenta.

Tabla 4.16 Ordenamiento ascendente por cada instancia real en función del tiempo de ejecución (casos de dominancia)

Instancia	***n***	***d***	***k***	**Ordenamiento ascendente del tiempo de ejecución (casos de dominancia)**
R1	4177	7	200	Pattern Reduction < Enhanced K-means < Early Classification
R1	4177	7	400	Enhanced K-means < Pattern Reduction < Early Classification
R1	4177	7	600	Enhanced K-means < Pattern Reduction < Early Classification
R1	4177	7	800	Enhanced K-means < Pattern Reduction < Early Classification
R1	4177	7	1000	Enhanced K-means < Pattern Reduction < Early Classification
R2	1030	8	100	Enhanced K-means < Pattern Reduction < Early Classification
R3	150	3	3	Pattern Reduction < Early Classification < Enhanced K-means
R3	150	3	5	Pattern Reduction < Early Classification < Enhanced K-means
R3	150	3	10	Pattern Reduction < Enhanced K-means < Early Classification
R3	150	3	20	Pattern Reduction < Enhanced K-means < Early Classification
R3	150	3	30	Enhanced K-means < Pattern Reduction < Early Classification
R3	150	3	40	Enhanced K-means < Pattern Reduction < Early Classification
R3	150	3	50	Enhanced K-means < Pattern Reduction < Early Classification
R4	20000	16	40	Pattern Reduction < Early Classification < Enhanced K-means
R4	20000	16	50	Pattern Reduction < Early Classification < Enhanced K-means
R4	20000	16	60	Pattern Reduction < Early Classification < Enhanced K-means
R4	20000	16	70	Pattern Reduction < Early Classification < Enhanced K-means
R4	20000	16	80	Pattern Reduction < Early Classification < Enhanced K-means
R4	20000	16	90	Pattern Reduction < Early Classification < Enhanced K-means
R4	20000	16	100	Pattern Reduction < Early Classification < Enhanced K-means
R5	657308	2	400	Pattern Reduction < Enhanced K-means < Early Classification
R6	245057	3	100	Pattern Reduction < Early Classification < Enhanced K-means
R7	600	60	6	Pattern Reduction < Early Classification < Enhanced K-means
R8	65740	15	20	Pattern Reduction < Early Classification < Enhanced K-means
R8	65740	15	40	Pattern Reduction < Early Classification < Enhanced K-means
R8	65740	15	60	Pattern Reduction < Early Classification < Enhanced K-means
R8	65740	15	80	Pattern Reduction < Enhanced K-means < Early Classification
R8	65740	15	100	Pattern Reduction < Enhanced K-means < Early Classification
R8	65740	15	120	Pattern Reduction < Enhanced K-means < Early Classification
R8	65740	15	140	Pattern Reduction < Enhanced K-means < Early Classification
R8	65740	15	160	Pattern Reduction < Enhanced K-means < Early Classification

Tabla 4.17 Ordenamiento ascendente por cada instancia sintética en función del tiempo de ejecución (casos de dominancia)

Instancia	***n***	***d***	***k***	**Ordenamiento ascendente del tiempo de ejecución (casos de dominancia)**
S1	2500	2	50	Early Classification < Pattern Reduction < Enhanced K-means
S1	2500	2	100	Enhanced K-means < Pattern Reduction < Early Classification
S1	2500	2	200	Enhanced K-means < Pattern Reduction < Early Classification
S1	25000	2	400	Enhanced K-means < Pattern Reduction < Early Classification
S1	25000	2	800	Enhanced K-means < Pattern Reduction < Early Classification
S2	10000	2	50	Early Classification < Pattern Reduction < Enhanced K-means
S2	10000	2	100	Early Classification < Pattern Reduction < Enhanced K-means
S2	10000	2	200	Early Classification < Pattern Reduction < Enhanced K-means
S2	10000	2	400	Enhanced K-means < Pattern Reduction < Early Classification
S2	10000	2	800	Enhanced K-means < Pattern Reduction < Early Classification
S3	20000	2	50	Early Classification < Pattern Reduction < Enhanced K-means
S3	20000	2	100	Early Classification < Pattern Reduction < Enhanced K-means
S3	20000	2	200	Early Classification < Pattern Reduction < Enhanced K-means
S3	20000	2	400	Early Classification < Pattern Reduction < Enhanced K-means
S3	20000	2	800	Enhanced K-means < Pattern Reduction < Early Classification
S4	40000	2	50	Early Classification < Pattern Reduction < Enhanced K-means
S4	40000	2	100	Early Classification < Pattern Reduction < Enhanced K-means
S4	40000	2	200	Early Classification < Pattern Reduction < Enhanced K-means
S4	40000	2	400	Early Classification < Pattern Reduction < Enhanced K-means
S4	40000	2	800	Early Classification < Pattern Reduction < Enhanced K-means
S5	6000	2	50	Early Classification < Pattern Reduction < Enhanced K-means
S6	6000	10	50	Pattern Reduction < Early Classification < Enhanced K-means
S7	6000	25	50	Pattern Reduction < Enhanced K-means < Early Classification
S8	6000	50	50	Pattern Reduction < Enhanced K-means < Early Classification
S9	6000	100	50	Early Classification < Pattern Reduction < Enhanced K-means
S10	6000	250	50	Pattern Reduction < Enhanced K-means < Early Classification
S11	6000	500	50	Pattern Reduction < Enhanced K-means < Early Classification
S12	6000	1000	50	Pattern Reduction < Enhanced K-means < Early Classification
S13	60000	2	50	Early Classification < Pattern Reduction < Enhanced K-means
S14	600000	2	50	Early Classification < Pattern Reduction < Enhanced K-means

4.8 Conclusiones del análisis comparativo

En la Tabla 4.18 se presenta un cuadro comparativo de las mejoras más relevantes del algoritmo K-means seleccionadas, en el cual se puntualizan las observaciones más sobresalientes al resolver las instancias de prueba contempladas en el caso de estudio.

Tabla 4.18 Cuadro comparativo de mejoras más relevantes al algoritmo K-means

Algoritmo / **Característica**	**Early Classification**	**Enhanced K-means**	**Pattern Reduction**
Parámetro que reduce	n	k	n
Técnica de optimización	Descartar objetos de futuros cálculos mediante la construcción de un umbral y un índice de equidistancia, donde los objetos cuyo índice de equidistancia es mayor al umbral de equidistancia son discriminados de cálculos futuros.	Guarda la distancia de cada objeto a su centroide más cercano y en la siguiente iteración sólo calcula la distancia hacia el mismo centroide. Si su distancia actual es menor o igual a la distancia previa, se evitan los demás cálculos.	Mediante el cálculo de la media, comprime los objetos más cercanos en un patrón representativo, de manera que se evitan cálculos redundantes.
Máximo porcentaje de reducción de tiempo	97.82%	83.74%	94.22%
Mínimo porcentaje de reducción de tiempo	3.44%	19.23%	40.37%
Máximo porcentaje de pérdida de calidad	-4.46	-1.93%	-24.97
Mínimo porcentaje de pérdida de calidad	0%	0%	-0.02%
Casos de dominancia respecto a la menor pérdida de calidad	28	33	0

Tabla 4.18 Cuadro comparativo de mejoras más relevantes al algoritmo K-means (continuación)

Algoritmo / **Característica**	**Early Classification**	**Enhanced K-means**	**Pattern Reduction**
Casos de dominancia respecto a la mayor reducción de tiempo	17	15	29
Ventajas	Este algoritmo presenta buenos resultados en la mayoría de los casos, sin embargo, presenta ventaja al trabajar con grandes cantidades de objetos.	Este algoritmo presenta buenos resultados para la mayoría de los casos, de forma particular, sus resultados se favorecen al trabajar con mayores valores de k.	Este algoritmo presenta buenos resultados sobre todo en gran número de dimensiones.
Desventajas	Para los casos de prueba, este algoritmo presentó una desventaja cuando el número de grupos se incrementa de tal modo que se aproxima al número de objetos.	En los casos de prueba, este algoritmo presentó una desventaja cuando se tiene un gran número de objetos agrupados en pocos grupos.	En los casos de prueba, este algoritmo se ve afectado cuando los datos no presentan una distribución normal.

Capítulo

5

Conclusiones y trabajos futuros

"No hay mejor medida de lo que una persona es que lo que hace cuando tiene completa libertad de elegir"

William Buelger (1934)

En este capítulo se presentan las aportaciones y conclusiones a las que se llegaron en el desarrollo de esta investigación. También, se sugieren algunos trabajos futuros para dar continuidad a esta línea de investigación.

5.1 Conclusiones

Con el desarrollo de esta investigación se muestra que es factible realizar un análisis comparativo de la mejora *Early Classification* y las dos mejoras más relevantes del algoritmo K-means en su fase de clasificación mediante métodos experimentales y un mecanismo de comparación de algoritmos.

Las principales aportaciones de esta investigación son:

1) Selección de las mejoras más relevantes al algoritmo K-means en su fase de clasificación.
2) Adecuación y uso de la metodología de McGeoch para el análisis comparativo de algoritmos.

Con relación a la selección de las dos mejoras más relevantes se utilizaron principios de revisión sistemática, recuperando 39 trabajos que presentan mejoras al algoritmo K-means en su fase de clasificación. Se realizó un ordenamiento descendente de los 10 artículos más citados en Scopus y Google Académico y mediante el análisis de dichos trabajos, se seleccionaron las mejoras *Enhanced K-means* y *Pattern Reduction.*

Respecto a la metodología para la comparación de algoritmos, se validó mediante su aplicación a un caso de estudio, donde se compararon las tres mejoras más relevantes del algoritmo K-means en su fase de clasificación y para ello se utilizó un enfoque experimental y rigor estadístico. Las principales conclusiones se describen a continuación.

a) En comparación con el algoritmo K-means estándar, la mejora *Enhanced K-means* presentó pérdidas de calidad menores al 2%, *Early Classification* obtuvo pérdidas de calidad menores al 5% y *Pattern Reduction* presentó pérdidas de calidad de hasta 24.97%. Por otra parte, en términos de eficiencia, la mejora *Enhanced K-means* presentó reducciones de tiempo de ejecución de hasta 83.74 %, *Early Classification* presentó una mejora en la eficiencia de hasta 97.82% y *Pattern Reduction* presentó reducciones de tiempo de hasta 94.22%.

b) Se identificaron 33 casos en los que la mejora *Enhanced K-means* presentó mejores resultados en términos de calidad. Los casos más representativos se obtuvieron al resolver las instancias R4, R5, R6, R8, S2, S3, S4, S13 y S14 (ver Tablas 4.2 y 4.3). Mediante la observación de estos casos se determinó su uso para instancias similares a las mencionadas anteriormente. Sus resultados son favorables, presentando una pérdida de calidad máxima de 1.93% y reducciones de tiempo desde 40% hasta 83.74%.

c) Se identificaron 28 casos en los que la mejora *Early Classification* presentó mejores resultados en términos de calidad. Los casos más representativos se obtuvieron al resolver las instancias R1, R3, R7, R8, S1 y S2 (ver Tablas 4.2 y 4.3). Mediante la observación de estos casos se determinó que el uso de esta mejora se recomienda para la solución de instancias con características similares a las mencionadas anteriormente. Se observaron casos como R1 con k=200, 400, 600, 800 y 1000 grupos, donde no se presentó pérdida de calidad pero se obtuvieron reducciones de tiempo menores al 15%, sin embargo, en otros casos como en S14, se obtuvieron reducciones de tiempo de ejecución de hasta 97.8 % y una pérdida de calidad de tan sólo 3%.

d) Se identificaron 29 casos en los que la mejora *Pattern Reduction* presentó mejores resultados en términos de reducción de tiempo, sin embargo, ésta presentó una pérdida de calidad mínima de 5% y máxima de hasta 24.97%. Mediante el análisis de los casos se identificó que debido al enfoque de solución el algoritmo presenta desventajas al resolver instancias con distribución no uniforme o con datos dispersos como en las instancias R5 y R6 (ver Tabla 4.2). Por otra parte, se observó un buen desempeño en la solución de las instancias S6, S7, S8, S9, S10, S11 y S12 (ver Tabla 4.3), por lo cual se recomienda su aplicación para la solución de instancias con gran número de dimensiones y una distribución normal.

5.2 Trabajos futuros

Con base en las observaciones obtenidas en esta tesis, a continuación se sugiere el desarrollo de dos trabajos futuros:

a) En esta investigación se utilizaron principios de revisión sistemática con reglas específicas de exclusión e inclusión para este estudio, sin embargo, se observó que existe una amplia literatura sobre la aplicación y mejora continua del algoritmo K-means. Por tal motivo, se propone desarrollar un marco de referencia que clasifique los trabajos sobre el algoritmo K-means con el objetivo de facilitar la revisión del estado del arte.

b) Durante el proceso de selección de algoritmos se observó que existen trabajos recientes que aún no cuentan con número de citas pero presentan resultados prometedores. En este sentido, se propone tomar como referencia los mejores resultados de este estudio y analizar los nuevos enfoques, con el objetivo de determinar nuevas mejoras relevantes y ampliar esta investigación.

Referencias

[1] C. H. Chen, S. D. Wu y L. Dai, "Ordinal comparison of heuristic algorithms ousing stochastic optimization," *IEEE Transactions on Robotics and Automation*, Vol. 15, No.1, pp. 44–56, 1999.

[2] D. H. Wolpert y W. G. Macready, "No free lunch theorems for optimization," I*EEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, pp. 67–82, 1997.

[3] C. C. McGeoch, *A Guide to Experimental Algorithmics*. Nueva York, Estados Unidos de América: Cambridge University Press, 2012.

[4] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," *Memorias de 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967.

[5] P. Drineas, A. Frieze, R. Kannan, S. Vempala y V. Vinay, "Clustering large graphs via the singular value decomposition," *Machine Learning*, Vol. 56, No. 1, pp. 9–33, 2004.

[6] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, Vol. 31, No. 8, pp. 651–666, 2010.

[7] J. Wu, *Advances in K-means Clustering: A Data Mining Thinking.* Springer Theses Recognizing Outstanding Ph.D. Research, 2012.

[8] X. Wu *et al.*, "Top 10 algorithms in Data Mining," *Knowledge and Information Systems,* Vol. 14, No. 1, pp. 1-37, 2007.

[9] J. Pérez *et al.*, "Mejora al algoritmo de agrupamiento K-means mediante un nuevo criterio de convergencia y su aplicación a Bases de Datos poblacionales de Cáncer," *Memorias del 2do Taller Latino Iberoamericano de Investigación de Operaciones*, Acapulco, Guerrero, México, 2007.

[10] C. Tsai, C. Yang y M. Chiang, "A time efficient Pattern Reduction algorithm for K-means based clustering," *Memorias de International Conference on Systems, Man and Cybernetics*, pp. 504–509, 2007.

[11] J. Z. C. Lai y. C. Liaw, "Improvement of the K-means clustering Filtering algorithm," *Pattern Recognition*, Vol. 41, No.12, pp. 3677–3681, 2008.

[12] N. Slonim, E. Aharoni y K. Crammer, "Hartigan's K-means versus Lloyd's K-means: Is it time for a change?," *Memorias de 23th International Joint Conference on Artificial Intelligence*, pp. 1677–1684, 2013.

[13] A. Mexicano, "Desarrollo de una metodología para la elección de atributos y generación de indicadores para la aplicación de Minería de Datos a una Base de Datos real de registros de Cáncer de base poblacional," Tesis de Maestría, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, México, 2007.

[14] M. A. Barrón, "Desarrollo de un prototipo para la aplicación de técnicas de Minería de Datos sobre una Base de Datos real de base poblacional de Cáncer," Tesis de Maestría, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, México, 2008.

[15] M. del R. Boone, "Identificación de regiones con altas tasas de incidencia de Cáncer mediante la integración y uso de técnicas de Minería de Datos: Almacenes de Datos, Agrupamiento y Sistemas de Información Geográficos," Tesis de Maestría, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, México, 2011.

[16] A. Mexicano, "Caracterización de conjuntos de instancias difíciles del problema de Bin Packing orientada a la mejora de algoritmos metaheurísticos mediante el uso de técnicas de Minería de Datos," Tesis Doctoral, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, México, 2012.

[17] R. I. Basave, "Mejoramiento de la eficiencia y eficacia del algoritmo de agrupamiento K-means mediante una nueva condición de convergencia," Tesis de Maestría, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, México, 2005.

[18] A. Moreno, "Mejora del algoritmo K-means incrementando su eficiencia en la fase de clasificación.," Tesis de Maestría, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, México, 2013.

[19] J. Pérez, C. E. Pires, L. Balby, A. Mexicano y M. A. Hidalgo, "Early Classification: A new heuristic to improve the classification step of K-means," *Journal of Information and Data Management,* Vol. 4, No. 2, pp. 1–10, 2013.

[20] V. López, "Incremento de la eficiencia del algoritmo K-means mediante la mejora de la heurística Early Classification," Tesis de Maestría, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, México, 2015.

[21] K. Sörensen, "Metaheuristics the metaphor exposed," *International Transactions in Operational Research*, Vol. 22, No. 1, pp. 3–18, 2015.

[22] R. Johnsonbaugh, *Matemáticas Discretas,* 6ª Edición. Edo. de México, México: Prentice Hall, 2005.

[23] K. H. Rosen, *Matemática Discreta y sus Aplicaciones,* 5ª Edición. Madrid, España: McGraw Hill, 2004.

[24] S. Lipschutz y M. Lipson, *Theory and Problems of Discrete Mathematics,* 3ª Edición. Nueva York, Estados Unidos de América: McGraw Hill, 2007.

[25] P. N. Tan, M. Steinbach y V. Kumar, *Introduction to Data Mining*, Massachusetts, Estados Unidos de América: Addison Wesley, 2006.

[26] H. H. Bock, "Origins and extensions of the K-means algorithm in cluster analysis," *Journal Electronique d'Histoire des Probabilités et de la Statistique*, Vol. 4, No. 2, pp. 1–18, 2008.

[27] L. Morissette y S. Chartier, "The K-means clustering technique: General considerations and implementation in Mathematica," *Tutorials in Quantitative Methods for Psychology*, Vol. 9, No. 1, pp. 15–24, 2013.

[28] J. A. Hartigan y M. A. Wong, "A K-means clustering algorithm," *Journal of the Royal Statistical Society*, Vol. 28, No. 1, pp. 100–108, 1979.

[29] S. P. Lloyd, "Least squares quantization in PCM," *Transactions on Information Theory*, Vol. 28, No. 2, pp. 129–137, 1982.

[30] M. Marrón, M. A. Sotelo y J. C. García, "Comparing improved versions of K-means and Subtractive Clustering in a Tracking Application," *Memorias de 11th International Conference on Computer Aided Systems Theory,* pp. 717-724, 2007.

[31] G. A. Wilkin y X. Huang, "A practical comparison of two K-means clustering algorithms.," *BMC Bioinformatics*, Vol. 9, No. 6, pp. 1-5, 2008.

[32] D. Qiu, "A comparative study of the K-means algorithm and the Normal Mixture Model for clustering: Bivariate homoscedastic case," *Journal of Statistical Planning and Inference*, Vol. 140, No. 7, pp. 1701–1711, 2010.

[33] M. E. Celebi, H. A. Kingravi y P. A. Vela, "A comparative study of efficient initialization methods for the K-means clustering algorithm," *Expert Systems with Applications Journal*, Vol. 40, No. 1, pp. 200–210, 2013.

[34] M. Zaït y H. Messatfa, "A comparative study of clustering methods," *Future Generation Computer Systems*, Vol. 13, No. 2, pp. 149–159, 1997.

[35] R. L. Rardin y R. Uzsoy, "Experimental evaluation of heuristic optimization algorithms : A tutorial," *Journal of Heuristics*, Vol. 7, No. 3, pp. 261–304, 2001.

[36] R. Bala, S. Sikka y J. Singh, "A Comparative analysis of clustering algorithms," *International Journal of Computer Applications*, Vol. 100, No. 15, pp. 35–40, 2014.

[37] C. Valdivieso, R. Valdivieso y O. Valdivieso, "Determinación del tamaño muestral mediante el uso de árboles de decisión," *UPB Investigación y Desarrollo*, Vol. 11, pp. 148–176, 2011.

[38] I. Ferreira, G. Urrútia y P. Alonso, "Revisiones sistemáticas y meta análisis: bases conceptuales e interpretación," *Revista Española de Cardiología,* Vol. 64, No. 8, pp. 688–696, 2011.

[39] C. Fuentelsaz, "Cálculo del tamaño de la muestra," *Matronas Profesión*, Vol. 5, No. 18, pp. 5–13, 2004.

[40] M. Gómez, C. Danglot y L. Vega, "Cómo seleccionar una prueba estadística: Primera de dos partes," *Revista Mexicana de Pediatría*, Vol. 80, No. 1, pp. 30–34, 2013.

[41] L. de la Torre, "Teoría del Muestreo," Departamento de Ingeniería Industrial, Instituto Tecnológico de Chihuahua, Reporte Técnico, 2003.

[42] M. L. Berenson y D. M. Levine, *Estadística Básica en Administración, Conceptos y Aplicaciones,* 6ª Edición. Edo. de México, México: Prentice Hall, 1996.

[43] J. Gorgas, N. Cardiel y J. Zamorano, *Estadística Básica para Estudiantes de Ciencias*. Madrid, España: Universidad Complutense de Madrid, 2011.

[44] M. F. Triola, *Estadística,* 9ª Edición. Edo. de México, México: Person Educación, 2004.

[45] J. A. García, A. Reding y J. C. López, "Cálculo del tamaño de la muestra en investigación en educación médica," *Investigación en educación médica*, Vol. 2, No 8, pp. 217-224, 2013.

[46] C. Pérez, *Muestreo estadístico, Conceptos y Problemas Resueltos*. Madrid, España: Pearson Prentice Hall, 2005.

[47] W. Mendenhall, R. J. Beaver y B. M. Beaver. *Introducción a la Probabilidad y Estadística, 13ª Edición*. Edo. de México, México: CENGAGE Learning, 2010.

[48] I. Rodriguez, A. Canosa, M. Mucientes y A. Bugarín, "STAC: A web platform for the comparison of algorithms using statistical test," *Memorias de International Conference on Fuzzy Systems*, Istanbul, Turkey, pp. 1–8, 2015.

[49] M. Cárdenas y H. Arancibia, "Potencia estadística y cálculo del tamaño del efecto en G*Power: complementos a las pruebas de significación estadística y su aplicación en Psicología," *Salud y Sociedad*, Vol. 5, No. 2, pp. 210–224, 2014.

[50] S. García, D. Molina, M. Lozan y F. Herrera, "Un estudio experimental sobre el uso de test no paramétricos para analizar el comportamiento de los algoritmos evolutivos en problemas de optimización," En *Memorias del Congreso Español sobre Metaheurísticas, algoritmos evolutivos y bioinspirados*, pp. 275–285, 2007.

[51] R. Coe y C. Merino, "Magnitud del efecto: Una guía para investigadores y usuarios," *Revista de Psicología de la PUCP*, Vol. 21, No. 1, pp. 145–177, 2003.

[52] M. Gómez, C. Danglot y L. Vega, "Cómo seleccionar una prueba estadística: Segunda parte," *Revista Mexicana de Pediatría*, Vol. 80, No. 2, pp. 81–85, 2013

[53] R. Shier, Mathematics Learning Support Centre, "Statistics: The Wilcoxon signed rank sum test," Fecha de consulta: Mayo 2016. [En línea]. Disponible en: http://www.lboro.ac.uk/media/wwwlboroacuk/content/mlsc/downloads/2.2_wsrt.pdf.

[54] C. Peng y X. Guiqiong, "A brief study on clustering methods based on the K-means algorithm," *Memorias de International Conference on E-Business and E-Government,* Shanghai, pp. 1–5, 2011.

[55] M. Gómez, *Elementos de estadística descriptiva*. Costa Rica: EUNED, 1998.

[56] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman y A. Wu, "An efficient K-means clustering algorithm: Analysis and implementation," *Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 7, pp. 881–892, 2002.

[57] C. Elkan, "Using the Triangle Inequality to accelerate K-means," *Memorias de 20th International Conference on Machine Learning,* pp. 147–153, 2003.

[58] A. Fahim, A. Salem, F. Torkey y M. Ramadan, "An efficient Enhanced K-means clustering algorithm," *Journal of Zhejiang University*, Vol. 7, No. 10, pp. 1626–1633, 2006.

[59] J. Z. C. Lai, T.-J. Huang y Y. C. Liaw, "A Fast-means clustering algorithm using cluster center displacement," *Pattern Recognition*, Vol. 42, No. 11, pp. 2551–2556, 2009.

[60] J. Z. C. Lai y T. J. Huang, "Fast global K-means clustering using cluster membership and inequality," *Pattern Recognition*, Vol. 43, No. 5, pp. 1954–1963, 2010.

[61] M.-C. Chiang, C.-W. Tsai y C. S. Yang, "A time-efficient Pattern Reduction algorithm for K-means clustering," *Information Sciences,* Vol. 181, No. 4, pp. 716–731, 2011.

[62] J. Wang, J. Wang, Q. Ke, G. Zeng y S. Li, "Fast approximate K-means via cluster closures," *Memorias de Conference on Computer Vision and Pattern Recognition*, pp. 3037–3044, 2012.

[63] V. C. Osamor, E. F. Adebiyi, J. O. Oyelade y S. Doumbia, "Reducing the time requirement of K-means algorithm," *PLoS ONE*, Vol. 7, No. 12, pp. 1-10, 2012.

[64] S.-S. Lee y J. C. Lin, "An accelerated K-means clustering algorithm using selection and erasure rules," *Journal of Zhejiang University*, Vol. 13, No. 10, pp. 761–768, 2012.

[65] K. Bache y M. Lichman, "UCI Machine Learning Repository," Fecha de consulta: Septiembre 2015. [En línea]. Disponible en: http://archive.ics.uci.edu/ml.

[66] University of Eastern Finland, "Clustering datasets," Fecha de consulta: Septiembre 2015. [En línea]. Disponible en: http://cs.joensuu.fi/sipu/datasets.

[67] Flickr, "Photo repository," Fecha de consulta: Septiembre 2015. [En línea]. Disponible: http://www.flickr.com/map.

[68] J. Pérez, R. Pazos, L. Cruz, G. Reyes, R. Basave y H. Fraire, "Improving the efficiency and efficacy of the K-means clustering algorithm through a new convergence condition," *Memorias de International Conference on Computational Science and Its Applications*, pp. 674–682, 2007.

[69] J. Pérez *et al.*, "Improvement to the K-means algorithm through a heuristics based on a Bee Honeycomb structure," *Journal of Network and Innovative Computing*, Vol. 1, No. 1, pp. 119–125, 2013.

Anexo

A

Análisis de las mejoras más relevantes del algoritmo K-means

En este Anexo se realiza un análisis del funcionamiento y especificaciones de las 3 mejoras más relevantes del algoritmo K-means seleccionadas en este trabajo.

A-1. An efficient Enhanced K-means clustering algorithm

Dado que el algoritmo K-means es un método iterativo para dividir un conjunto de objetos en diferentes grupos en función al centroide más cercano, su complejidad computacional aumenta a medida que se incrementa el tamaño (n) de las instancias a resolver. Mediante esta observación, los autores se formulan la pregunta -*¿Por qué no beneficiarse de cada iteración del algoritmo?* - Con este enfoque, en este trabajo se propone una mejora al algoritmo K-means denominada *Enhanced K-means*, donde la idea es, guardar la distancia de cada objeto al centroide más cercano en cada iteración y posteriormente realizar el cálculo de distancia de cada objeto a su centroide más cercano en la iteración previa, de modo que, si la distancia en la iteración actual es menor o igual a la distancia previamente almacenada, el objeto no cambia de grupo y sólo se actualiza el valor de su distancia. Si esta condición se cumple, el algoritmo se beneficia al reducir su costo computacional evitando los cálculos de distancia de cada objeto a todos los centroides.

Los dos métodos propuestos por los autores son *distance*() y *distance_new*(), éstos, se muestran en su forma e idioma original en la Figura A.1 y la Figura A.2, respectivamente.

El método *distance*() realiza el proceso del algoritmo K-means estándar, el cual, consiste en calcular la distancia de cada objeto a todos los centroides y asignar cada objeto al grupo cuyo centroide es el más cercano. El enfoque que se introduce es la creación de dos estructuras de datos, en las cuales, se almacena la distancia de cada objeto al centroide más cercano en cada iteración y el índice de dicho centroide (líneas 8 y 9 de la Figura A.1).

```
Function distance()
//assign each point to its nearest cluster
1 For i = 1 to n
2   For j = 1 to k
3    Compute squared Euclidean distance
          d²(x_i, m_j);
4   endfor
5         Find the closest centroid m_j to x_i;
6         m_j = m_j + x_i;  n_j = n_j + 1;
7         MSE = MSE + d²(x_i, m_j);
8         Clusterid[i] =number of the closest centroid;
9         Pointdis[i] =Euclidean distance to the closest centroid;
10 endfor
11 For j = 1 to k
12    m_j = m_j/n_j;
13 endfor
```

Figura A.1 Función *distance*() propuesta en *Enhanced K-means*

El método *distance_new*() calcula la nueva distancia de cada objeto al centroide del grupo al cual pertenecía en la iteración anterior y compara su distancia en la iteración anterior y la actual (línea 1 de la Figura A.2). Si la distancia en la iteración actual es menor o igual a la distancia previamente almacenada, el objeto permanece en el mismo grupo y no hay necesidad de calcular las distancias a los otros k-1 centroides,

de modo contrario, se realizan todos los cálculos de distancia hasta encontrar el nuevo centroide más cercano para cada objeto.

```
Function distance_new()
//assign each point to its nearest cluster
1 For i = 1 to n
      Compute squared Euclidean distance
        d²(x_i, Clusterid[i]);
      If (d²(x_i, Clusterid[i]) <= Pointdis[i])
        Point stay in its cluster;
2     Else
3       For j = 1 to k
4           Compute squared Euclidean distance
              d²(x_i, m_j);
5       endfor
6     Find the closest centroid m_j to x_i;
7     m_j = m_j + x_i;  n_j = n_j + 1;
8     MSE = MSE + d²(x_i, m_j );
9     Clusterid[i] =number of the closest centroid;
10    Pointdis[i] =Euclidean distance to the closest centroid;
11 endfor
12 For j = 1 to k
13     m_j = m_j/n_j;
14 endfor
```

Figura A.2 Función *distance_new*() propuesta en *Enhanced K-means*

Mientras los centroides sean diferentes entre una iteración y otra, la función *distance*() se implementa si la iteración es menor o igual a dos, de lo contrario se ejecuta la función *distance_new*().

A-1.1 Análisis del algoritmo *Enhanced K-means*

Mediante un análisis detallado de la mejora *Enhanced K-means* se obtuvo información relevante sobre su funcionamiento y detalles de la misma para su implementación

computacional. Con el objetivo de apoyar a los investigadores, a continuación se presentan dichas especificaciones y argumentan aspectos importantes de este trabajo.

a) Condición de inicio

Enhanced K-means recibe como parámetro de entrada un conjunto de objetos de tamaño n, el número de k grupos, el número de d-dimensiones de la instancia y un conjunto de k centroides iniciales seleccionados aleatoriamente de la instancia a agrupar.

En el inciso b se detallan los métodos propuestos en este trabajo, los cuales se inicializan de la siguiente manera: El primero, se ejecuta en las dos primeras iteraciones, mientras que el segundo, se ejecuta a partir de la tercera iteración y hasta su convergencia. Respecto a la convergencia, ésta se cumple si todos los centroides ya no cambian en dos iteraciones consecutivas.

b) Análisis de los métodos propuestos

De acuerdo con el análisis teórico y experimental de este trabajo, se encontraron errores de omisión en los pseudocódigos presentados por los autores. En el método *distance*() presentado en la Figura A.1 se encontraron dos errores, el primero, en la línea 4, donde se presenta el cierre de un ciclo desde j=1 hasta k, de manera que al seguir con el método, se desconoce el valor que tiene j para el resto de las asignaciones. El segundo en la línea 6 donde se expresa la sustitución del valor del centroide m_j.

La corrección del método *distance*() se presenta en la Figura A.3, de la siguiente manera: En la línea 4 se calcula la distancia Euclidiana al cuadrado entre el objeto x_i y el centroide m_j, en este proceso se realizan nk cálculos de distancia, las cuales son comparadas hasta encontrar la menor distancia de cada objeto x_i a un centroide m_j (líneas 5-8). En la línea 10, cada objeto es asignado al grupo j cuyo centroide m_j es el más cercano y se activa un contador que incrementa en uno cada vez que se asigna un objeto, esto, con la finalidad de realizar el cálculo del nuevo centroide m_j como se aprecia en la línea 17. La línea 12, consiste en añadir a la variable *ECM* (Error Cuadrático Medio, *MSE* en

inglés, *Mean Squared Error*) la mínima distancia de cada objeto x_i y el centroide m_j, este proceso es calculado con la expresión A-1.1 y representa la función objetivo del algoritmo, la cual es minimizar la sumatoria del error al cuadro (*SSE* por sus siglas en inglés, *Sum of Squared Errors*).

$$\frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{n_i}\|x_{ij}-m_i\|^2 \qquad \text{(A-1.1)}$$

Donde, n indica el número de objetos, k es número de grupos, n_i es el número de objetos en el grupo i y $\|x_{ij}-m_i\|^2$ expresa la norma Euclidiana al cuadrado del j-ésismo objeto del grupo i y el centroide m_i.

En las líneas 13 y 14 de la Figura A.3 se muestran las dos estructuras de datos que se agregan al algoritmo con la finalidad de guardar en índice j del centroide más cercano al objeto x_i y su distancia Euclidiana al cuadrado, respectivamente. Por último de la línea 16 a la línea 18 se realiza el cálculo del nuevo centroide m_j.

```
Función distancia()
1   Desde i = 1 hasta n
2       distancia = 0;
3       Desde j = 1 hasta k
4               Calcular la distancia Euclidiana al cuadrado
        de cada objeto a cada centroide
                dE = ||x_i - m_j ||^2;
5           Si distancia > dE Entonces
6               distancia = dE;
7               índice = j;
8           Fin_Si
9       Fin_Desde
10      Asignar el objeto x_i al grupo m_índice;
            grupo_m_índice = grupo_m_índice + x_i;
11          n_índice = n_índice + 1;
12              ECM = ECM + distancia;
13          índice_grupo[i] = índice;
14          distancia_menor[i] = distancia;
15  Fin_Desde
16  Desde j = 1 hasta k
17      m_j = grupo_m_índice/n_índice;
18  Fin_Desde
```

Figura A.3 Método *distancia*()

Las correcciones del método *distance_new*() se muestran en la Figura A.4 de la siguiente manera: de la línea 2 a la línea 5, se presenta la idea principal de esta mejora, donde, se calcula la distancia Euclidiana al cuadrado de cada objeto $x_i$al centroide m_j del grupo al cual pertenecía en la iteración anterior (línea 2). Si la nueva distancia calculada es menor o igual a la distancia almacenada en la iteración anterior, el objeto permanece en el mismo grupo y actualiza su nueva distancia (líneas 3-4). Cabe mencionar que, si esta condición se cumple, el algoritmo evita *k*-1 cálculos de distancia y realiza sólo uno, de modo que la complejidad computacional se minimiza de *nk* a 1. Por otro lado, si la condición no se cumple se realizan las líneas 6-20.

Función *nueva_distancia*()
1 **Desde** $i = 1$ **hasta** n
2 Calcular la distancia Euclidiana al cuadrado del objeto x_i a su centroide más cercano en la iteración anterior $dE = ||x_i - índice\_grupo[i]||^2$;
3 **Si** ($dE <= distancia\_menor[i]$) **Entonces**
4 El objeto x_i permanece en el grupo j y se actualiza su distancia $distancia\_menor[i] = dE$;
5 **Fin_Si**
6 **Sino**
7 $distancia = 0$;
8 **Desde** $j = 1$ **hasta** k
9 Calcular la distancia Euclidiana al cuadrado de cada objeto a cada centroide $dE = ||x_i - m_j||^2$;
10 **Si** $distancia > dE$ **Entonces**
11 $distancia = dE$;
12 $índice = j$;
13 **Fin_Si**
14 **Fin_Desde**
15 Asignar el objeto x_i al grupo $m_{índice}$; $grupo\_m_{índice} = grupo\_m_{índice} + x_i$;
16 $n_{índice} = n_{índice} + 1$;
17 $ECM = ECM + distancia$;
18 $índice\_grupo[i] = índice$;
19 $distancia\_menor[i] = distancia$;
20 **Fin_Sino**
20 **Fin_Desde**
21 **Desde** $j = 1$ **hasta** k
22 $m_j = grupo\_m_{índice}/n_{índice}$;
23 **Fin_Desde**

Figura A.4 Método *nueva_distancia*()

A-2. A time efficient Pattern Reduction algorithm for K-means clustering

Dado que, el algoritmo K-means estándar realiza el cálculo de distancia de cada objeto a todos los centroides en cada iteración, los autores proponen una mejora llamada *Pattern Reduction (PR)* con la finalidad de eliminar cálculos redundantes que contribuyen a un tiempo de procesamiento elevado. La idea es identificar los objetos con baja probabilidad de cambio de grupo mediante dos métodos, a saber: compresión y eliminación de patrones *(PCR)* y asignación de patrones y actualización de medias *(PAMU)*, los cuales se muestran en su forma e idioma original en las Figuras A.5 y A.6, respectivamente.

El método *PCR* (ver Figura A.5) calcula la media y la desviación estándar de las distancias de todos los objetos pertenecientes al grupo, las cuales se utilizan para localizar los objetos que se encuentran más cercanos al centroide. Estos objetos se eliminan de futuros cálculos y son comprimidos, de manera que la media de dichos objetos (patrón representativo) representa la información de éstos.

Procedure PCR
{
  if the porcentage of patterns removed < removal bound **do**
    for each cluster i **do**
      1. Compute the mean μ and standard deviation σ of the distances of all patterns in the cluster to their cluster center.
      2. Use the μ and σ computed above to locate and remove patterns near the cluster center.
      3. Compress the patterns removed into a new pattern and update M
    end
  end
}

Figura A.5 Proceso de compresión y eliminación de patrones (PCR)

Los objetos que no fueron comprimidos con el método PCR, son reasignados al centroide más cercano mediante el método *PAMU* y se actualizan los centroides como

la media de todos los objetos pertenecientes a cada grupo, tomando en cuenta la partición actual. El paso final de este método es el cálculo de la sumatoria del error al cuadrado (ver Figura A.6).

```
Procedure PAMU
{
   if the porcentage of patterns removed < removal baound do
      for each cluster i do
         Reassing each pattern not removal to the closest cluster center
      end
      for each cluster i do
         Compute the new mean C_i^l
      end
      Compute the new  SSE at iteration l
    end
}
```

Figura A.6 Proceso de asignación de patrones y actualización de medias (PAMU)

A-1.1 Análisis del algoritmo *Pattern Reduction*

Mediante un análisis detallado de la mejora *Pattern Reduction* se obtuvo información relevante sobre su funcionamiento y detalles de la misma para su implementación computacional. Con el objetivo de apoyar a los investigadores, a continuación se presentan dichas especificaciones y se argumentan aspectos importantes de este trabajo.

a) Condición de inicio

Pattern Reduction recibe como parámetros de entrada un conjunto de objetos de tamaño n, el número de k grupos, el número de d-dimensiones de la instancia y un conjunto de k centroides iniciales seleccionados aleatoriamente de la instancia a agrupar.

En el inciso b se detallan los métodos propuestos en este trabajo, los cuales se inicializan de la siguiente manera: En las primeras dos iteraciones se ejecuta el algoritmo K-means estándar, mientras que, a partir de la tercera iteración se

ejecuta primero el método PCR y después el método PAMU hasta su convergencia. Respecto a la convergencia, ésta se cumple si todos los centroides ya no cambian en dos iteraciones consecutivas. Para apoyar a los investigadores en la Figura A.7 se muestra dicho proceso.

```
1 Algoritmo Pattern Reduction
2    Generar los centroides iniciales como la mejor solución inicial;
3    Si iteración > 2 y objetos eliminados < 80% hacer
4        Ejecutar PCR y PAMU;
5    Fin_si
6    Sino
7        Asignar cada objeto al grupo cuyo centroide es el más cercano;
8        Actualizar los centroides;
9    Fin_Sino
10   Si los centroides no cambian en dos iteraciones consecutivas i hacer
11       Terminar;
12   Fin_Si
13   Sino
14    Regresar a la línea 3;
15   Fin_Sino
16 Fin_Algoritmo
```

Figura A.7 Implementación del algoritmo *PR* en K-means

b) Análisis de los métodos propuestos

De acuerdo con el análisis teórico y experimental de este trabajo, a continuación se presentan las especificaciones de cada método propuesto.

El método PCR (ver Figura A.8) inicia verificando que el número total de objetos eliminados no supere el 80% (línea 2). De la línea 3 a la línea 7 se muestra el proceso de comprensión y eliminación de patrones, el cual consiste en calcular la media y la desviación estándar de las distancias de todos los objetos pertenecientes al grupo i, esto, con el objetivo de formar un umbral que permita localizar los objetos más cercanos al grupo i, es decir, los objetos que se encuentren dentro del umbral son eliminados de futuros cálculos y comprimidos en un patrón representativo como la media de todos los objetos eliminados.

```
1  Método PCR
2     Si porcentaje de objetos eliminados < 80% hacer
3        Desde i = 1 hasta k hacer
4              Calcular la media μ y desviación estándar σ de las distancias de
                todos los objetos pertenecientes al grupo i;
5              Formar umbral como μ - σ para localizar y eliminar los objetos
                más cercanos al centroide del grupo i;
6              Comprimir los objetos eliminados en un patrón representante
                como la media de todos los objetos eliminados en el grupo i;
7        Fin_Desde
8     Fin_Si
8  Fin_Método
```

Figura A.8 Método de comprensión y eliminación de objetos

El método *PAMU* (ver Figura A.9), al igual que el método anterior, inicia verificando que el número total de objetos eliminados no supere el 80% (línea 2). Si existen objetos que no fueron comprimidos, éstos, son reasignados al grupo cuyo centroide es el más cercano (línea 4), seguidamente se calculan los nuevos centroides (líneas 6-8) y la sumatoria del error al cuadrado (línea 9).

```
1  Método PAMU
2     Si porcentaje de objetos eliminados < 80% hacer
3        Desde i = 1 hasta k hacer
4           Reasignar cada objeto no eliminado al grupo cuyo
            centroide es el más cercano;
5        Fin_Desde
6        Desde i = 1 hasta k hacer
7           Calcular los nuevos centroides;
8        Fin_Desde
9        Calcular la sumatoria de error al cuadrado;
10    Fin_Si
11 Fin_Método
```

Figura A.9 Método de asignación de objetos y actualización de centroides

Es importante mencionar que los dos métodos son alternados en cada iteración, primero PCR y luego PAMU. Si se cumple el 80% de objetos eliminados, el algoritmo continuará como el procedimiento del algoritmo K-means estándar hasta que los centroides ya no cambien en dos iteraciones consecutivas

A-3. Early Classification: A new heuristic to improve the classification step of K-means

Dado que, la complejidad del algoritmo K-means es *O(nkdl),* está claro que además del número de objetos, un factor que afecta en gran medida el costo computacional de K-means es el número de iteraciones que el algoritmo tiene que llevar a cabo. En este contexto, los autores proponen una mejora al algoritmo K-means denominada *Early Classification,* donde, la idea es, reducir el número de cálculos necesarios en la etapa de clasificación del algoritmo K-means sin pérdida significativa de la calidad, para esto, se identifican los objetos con baja probabilidad de cambio de grupo y son excluidos de futuros cálculos. Para llevar a cabo este proceso se introducen los conceptos de índice y umbral de equidistancia.

El índice de equidistancia (α_i) expresa la diferencia de las distancias de un objeto a sus dos centroides más cercanos, mientras que, el umbral de equidistancia (β_j) está definido por los dos desplazamientos mayores de los centroides.

La magnitud del umbral varía entre la última iteración y la actual, ya que está directamente relacionada con los desplazamientos de los centroides. El proceso de discriminación de objetos para futuros cálculos se representa en la Figura A.10 y consiste en marcar los elementos cuyo índice de equidistancia sea mayor al umbral de equidistancia en cada iteración.

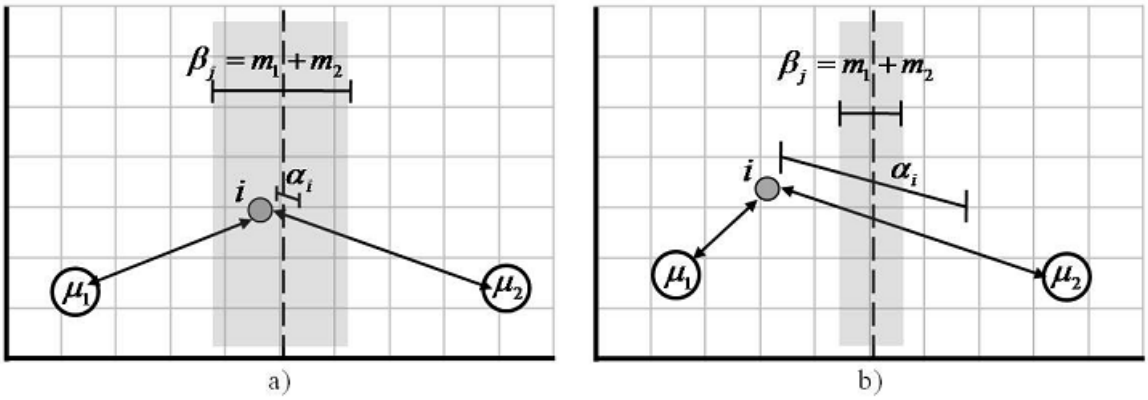


Figura A.10 Early Classification: a) alta probabilidad de cambio, b) baja probabilidad de cambio

A-1.1 Análisis del algoritmo *Early Classification*

Mediante un análisis detallado de la mejora *Early Classification* se obtuvo información relevante sobre su funcionamiento y detalles de la misma para su implementación computacional. Con el objetivo de apoyar a los investigadores, a continuación se presentan dichas especificaciones y se argumentan aspectos importantes de este trabajo.

a) Condición de inicio

Early Classification recibe como parámetros de entrada un conjunto de objetos de tamaño n, el número de k grupos, el número de d-dimensiones de la instancia y un conjunto de k centroides iniciales seleccionados aleatoriamente de la instancia a agrupar.

En el inciso b se detallan los métodos propuestos en este trabajo, los cuales se inicializan de la siguiente manera: En las primeras dos iteraciones se ejecuta el algoritmo K-means estándar, mientras que, a partir de la tercera iteración se ejecuta la mejora *Early Classification*. Respecto a la convergencia, ésta se cumple si todos los centroides ya no cambian en dos iteraciones consecutivas.

b) Análisis de los métodos propuestos

De acuerdo con el análisis teórico y experimental de este trabajo, a continuación se presentan las especificaciones de esta mejora.

Para apoyar a la comprensión de esta mejora se diseñó un pseudocódigo que muestra los métodos propuestos (ver Figura A.11). El algoritmo inicia con la generación de centroides iniciales (línea 2). De la línea 3 a la línea 16 se muestra el proceso de *Early Classification* el cual inicia calculando la distancia Euclidiana al cuadrado de cada objeto a todos los centroides y guardando para cada objeto sus dos centroides más cercanos (línea 6). En la línea 7 cada objeto es asignado al grupo j cuyo centroide es el más cercano.

Utilizando las distancias de los dos centroides más cercanos a cada objeto, se calcula el índice de equidistancia como la diferencia de dichas distancias (línea 8). En la línea 9 se realiza el cálculo de los nuevos centroides como la media de todos los objetos pertenecientes al grupo j, durante este proceso, se obtienen los desplazamientos de los centroides entre una iteración y otra, de los cuales se guardan los dos desplazamientos mayores. En la línea 10 se calcula el umbral de equidistancia como la sumatoria de los dos desplazamientos mayores de los centroides y en la línea 11 se introduce la condición de discriminación de objetos, de modo que, si el índice de equidistancia de cada objeto es mayor al umbral de equidistancia, éstos son eliminados de futuros cálculos.

```
1 Algoritmo Early Classification
2     Generar los centroides iniciales:
3         Si iteración > 2 hacer
4           Desde i = 1 hasta n hacer
5              Desde j = 1 hasta k hacer
6                  Calcular la distancia euclidiana al cuadrado del objeto x_i al centroide m_j y
                   Guardar la distancia de los dos centroides más cercanos para el objeto x_i;
7                  Asignar cada objeto x_i al grupo cuyo centroide m_j es el más cercano:
8                  Calcular índice de equidistancia como la diferencia de las distancias del objeto
                   x_i a sus dos centroides más cercanos
9                  Calcular los nuevos centroides y guardar los dos desplazamientos mayores;
10                 Calcular el umbral de equidistancia como la sumatoria de los dos
                   desplazamientos mayores de los centroides
11                 Si índice de equidistancia > umbral de equidistancia hacer
12                     Excluir objeto x_i de futuros cálculos;
13                 Fin_Si
14             Fin_Desde
15           Fin_Desde
16         Fin_si
17         Sino
18           Ejecutar K-means estándar;
19           Actualizar los centroides;
20         Fin_Sino
21         Si los centroides no cambian en dos iteraciones consecutivas i hacer
22             Terminar;
23         Fin_Si
24         Sino
25             Regresar a la línea 3;
26         Fin_Sino
27 Fin_Algoritmo
```

Figura A.11 Pseudocódigo de *Early Classification*

Anexo

B

Resultados experimentales

En este Anexo se muestran los resultados obtenidos en la experimentación. En la Tabla B.1 se muestran los resultados promedios de 30 ejecuciones para las instancias reales en términos de sumatoria del error al cuadrado (SSE) y tiempo de ejecución; mientras que en la Tabla B.2 se muestran los resultados de las instancias sintéticas.

Tabla B.1 Resultados experimentales con instancias reales en términos de SSE y Tiempo

Instancia	n	d	k	K-means Estándar		Early Classification		Enhanced K-means		Pattern Reduction	
				SSE	Tiempo	SSE	Tiempo	SSE	Tiempo	SSE	Tiempo
Abalone	4177	7	200	181.4	6273.2	181.9	3742.4	182.8	1487.6	190.3	1243.6
Abalone	4177	7	400	152.2	8598.5	152.3	6470.8	153.7	2084.3	157.8	2233.9
Abalone	4177	7	600	135.5	9871.9	135.5	8299.9	136.9	2556.5	139.7	3314.5
Abalone	4177	7	800	122.7	10869.2	122.7	9566.0	123.9	3066.3	126.1	3314.5
Abalone	4177	7	1000	112.4	12667.9	112.4	11096.1	113.4	3514.2	115.2	5030.0
Concrete	1030	8	100	42890.5	345.9	42941.8	243.4	43719.6	111.9	45062.6	135.7
Iris	150	3	3	95.4	0.4	95.6	0.2	95.5	0.3	100.9	0.2
Iris	150	3	5	73.1	1.0	73.9	0.5	73.1	0.6	77.5	0.3
Iris	150	3	10	54.7	1.8	55.0	1.0	55.3	0.9	57.5	0.7
Iris	150	3	20	41.0	3.2	41.0	2.1	41.4	1.4	42.4	1.4
Iris	150	3	30	33.3	4.5	33.3	3.3	33.7	1.9	34.7	2.0
Iris	150	3	40	27.9	5.6	27.9	4.3	28.2	2.3	29.1	2.7
Iris	150	3	50	23.8	5.6	23.8	4.9	24.0	2.6	24.5	3.3
Letter	20000	16	40	99377.9	54701.9	100434.0	6035.2	99504.2	16544.8	104510.6	5531.2

Tabla B.1 Resultados experimentales con instancias reales en términos de SSE y Tiempo (continuación)

Instancia	*n*	*d*	*k*	K-means Estándar		Early Classification		Enhanced K-means		Pattern Reduction	
				SSE	Tiempo	SSE	Tiempo	SSE	Tiempo	SSE	Tiempo
Letter	20000	16	50	95111.6	59343.3	96123.8	7719.7	95201.9	20950.5	99920.6	6642.5
Letter	20000	16	60	91456.3	70722.8	92430.6	9831.8	91659.4	22481.2	96426.0	8073.2
Letter	20000	16	70	88550.6	85177.8	89394.1	11598.6	88667.2	22966.9	93393.4	9588.6
Letter	20000	16	80	86296.4	101305.4	87111.4	13233.4	86435.3	25105.9	90886.0	9885.0
Letter	20000	16	90	84185.3	95564.2	84856.0	15368.2	84362.8	24077.2	88646.7	11388.5
Letter	20000	16	100	82305.2	103365.2	82897.9	17330.3	82499.4	25943.1	86701.9	12725.8
New York	245057	3	100	1439.2	2015454	1501.4	539107	1460.0	423637	1781.6	213850
Skin	657308	2	100	1881762	401905.6	1965730	60332.2	1898027	99854.3	2351598	23224.7
uci-sc	600	60	50	23420.1	80.2	23431.5	42.3	23434.6	51.5	23930.4	38.1
Wind	6574	15	20	68738.0	6005.5	69135.8	770.9	68758.4	2367.2	70693.4	647.4
Wind	6574	15	40	62706.9	11422.3	63059.1	1884.2	62790.1	3382.2	64664.0	1305.2
Wind	6574	15	60	59685.6	15584.6	59917.5	3268.4	59812.51	3888.1	61496.8	1778.3
Wind	6574	15	80	57611.0	16713.0	57771.5	4671.7	57747.2	4254.5	59186.3	2347.6
Wind	6574	15	100	55996.0	18612.0	56104.5	6092.8	56128.6	4785.8	57496.0	3047.2
Wind	6574	15	120	54683.6	19494.2	54763.9	7675.8	54825.6	5124.6	56118.5	3483.6
Wind	6574	15	140	53594.1	21278.2	53649.0	8988.8	53757.9	5328.9	54981.2	3907.4
Wind	6574	15	160	52609.5	24923.0	52659.6	10964.1	52814.1	5708.1	54015.6	4346.9

Tabla B.2 Resultados experimentales con instancias sintéticas en términos de SSE y Tiempo

Instancia	*n*	*d*	*k*	K-means Estándar		Early Classification		Enhanced K-means		Pattern Reduction	
				SSE	Tiempo	SSE	Tiempo	SSE	Tiempo	SSE	Tiempo
2500	2500	2	50	6806.7	368.6	6915.5	72.1	6865.5	100.9	7188.9	77.8
2500	2500	2	100	4852.4	578.5	4890.5	152.8	4910.4	133.9	5065.6	137.5
2500	2500	2	200	3451.6	716.6	3457.6	322.8	3499.4	183.1	3563.3	248.7
2500	2500	2	400	2444.7	930.3	2445.2	623.3	2472.3	274.4	2490.8	438.8
2500	2500	2	800	1704.7	1273.1	1704.7	1124.1	1713.4	443.9	1713.6	612.3
10000	10000	2	50	54177.4	3958.1	55614.5	284.9	54355.8	1009.7	58240.6	445.8
10000	10000	2	100	38363.6	6523.6	39214.6	595.4	38625.3	1361.2	40978.8	802.2
10000	10000	2	200	27255.4	8372.7	27624.1	1301.7	27485.7	1678.6	28756.9	1426.5
10000	10000	2	400	19367.3	11269.9	19484.4	2888.6	19587.1	2189.0	20246.4	2404.4
10000	10000	2	800	13784.4	13430.8	13796.5	6244.1	13958.5	2971.4	14216.0	4203.4
20000	20000	2	50	153163.0	10552.9	157823.5	559.9	153591.5	2782.8	165343.1	1084.9
20000	20000	2	100	108233.0	20464.3	111046.7	1188.3	108719.5	4254.1	116204.5	2049.3
20000	20000	2	200	76673.5	28725.9	78142.9	2533.1	77105.2	5320.0	81451.8	3578.8
20000	20000	2	400	54414.5	40085.2	55031.9	5599.6	54840.1	6700.9	57293.5	6056.0
20000	20000	2	800	38690.2	48673.0	38866.0	12575.0	39108.0	8813.1	40323.7	10775.7
40000	40000	2	50	432573.2	25790.0	444106.1	1116.6	433020.0	8296.8	461749.5	2716.8
40000	40000	2	100	305172.1	58343.2	312266.7	2293.1	305787.8	13334.1	324437.6	5076.1
40000	40000	2	200	216011.0	96967.3	220729.4	5146.6	216771.3	17368.2	230194.7	9087.6
40000	40000	2	400	153156.3	131262.4	155575.8	10965.2	153905.4	21792.6	161927.2	15607.5
40000	40000	2	800	108735.8	170438.2	109837.4	24507.9	109545.3	27710.7	114426.1	27003.6
DSH1	6000	2	50	320.4	1579.4	330.0	172.2	323.6	404.2	343.8	251.8

Tabla B.2 Resultados experimentales con instancias sintéticas en términos de SSE y Tiempo (continuación)

Instancia	*n*	*d*	*k*	K-means Estándar		Early Classification		Enhanced K-means		Pattern Reduction	
				SSE	Tiempo	SSE	Tiempo	SSE	Tiempo	SSE	Tiempo
DSH2	6000	10	50	3781.7	9142.6	3803.2	1628.3	3787.2	2598.9	3879.8	1262.2
DSH3	6000	25	50	7503.5	18264.1	7508.9	6748.8	7510.1	5667.5	7590.6	3085.3
DSH4	6000	50	50	11345.4	27555.2	11345.6	16899.1	11350.4	9291.6	11426.0	5678.3
DSH5	6000	100	50	224.5	2556.4	229.2	374.2	227.1	519.5	239.8	436.5
DSH6	6000	250	50	26828.9	81986.1	26828.9	75572.72	26832.0	29632.6	26875.2	22475.1
DSH7	6000	500	50	38261.6	124854.3	38261.6	120560.2	38263.9	49131.4	38289.0	42308.6
DSH8	6000	1000	50	54312.8	206348.9	54312.8	195695.9	54314.1	81387.3	54324.8	75589.3
DSL1	60000	2	50	3231.3	46303.2	3331.8	1804.4	3234.0	15408.2	3491.3	4540.6
DSL2	600000	2	50	32400.7	973961.6	33395.6	21279.9	32407.3	448943.1	34808.1	96760.0

Respecto a los resultados obtenidos, se llevó a cabo un post-procesamiento de datos que consistió en la integración de datos de forma tabular obteniendo promedios de 30 ejecuciones y se calcularon los índices seleccionados para el estudio comparativo. En este sentido, en la Tabla B.3 se presentan los resultados promedio de 30 ejecuciones por cada instancia real añadiendo los índices de porcentaje de pérdida de calidad y porcentaje de reducción de tiempo. Por otra parte en la Tabla B.4 se presentan los resultados para instancias sintéticas.

Tabla B.3 Resultados experimentales con instancias reales en términos de porcentaje de pérdida de calidad y porcentaje de reducción de tiempo

				Early Classification		Enhanced K-means		Pattern Reduction	
Instancia	*n*	*d*	*k*	%Calidad	%Tiempo	%Calidad	%Tiempo	%Calidad	%Tiempo
Abalone	4177	7	200	-0.27	40.34	-0.79	76.29	-4.92	80.18
Abalone	4177	7	400	-0.08	24.75	-1.03	75.76	-3.70	74.02
Abalone	4177	7	600	-0.01	15.92	-1.03	74.10	-3.14	66.42
Abalone	4177	7	800	0.00	11.99	-0.98	71.79	-2.72	69.50
Abalone	4177	7	1000	0.00	12.41	-0.89	72.26	-2.54	60.29
Concrete	1030	8	100	-0.12	29.62	-1.93	67.65	-5.06	60.76
Iris	150	3	3	-0.25	33.91	-0.10	19.23	-5.72	52.47
Iris	150	3	5	-1.11	49.47	-0.05	35.69	-6.13	64.74
Iris	150	3	10	-0.51	42.62	-1.09	47.02	-5.12	59.58
Iris	150	3	20	-0.08	34.58	-0.97	54.47	-3.37	55.70
Iris	150	3	30	-0.05	27.80	-1.29	57.14	-4.35	54.41
Iris	150	3	40	-0.11	22.73	-1.20	57.98	-4.21	52.38
Iris	150	3	50	0.00	13.21	-0.85	53.72	-2.65	40.37
Letter R.	20000	16	40	-1.06	88.97	-0.13	69.75	-5.16	89.89
Letter R.	20000	16	50	-1.06	86.99	-0.09	64.70	-5.06	88.81

Tabla B.3 Resultados experimentales con instancias reales en términos de porcentaje de pérdida de calidad y porcentaje de reducción de tiempo (continuación)

				Early Classification		Enhanced K-means		Pattern Reduction	
Instancia	***n***	***d***	***k***	**%Calidad**	**%Tiempo**	**%Calidad**	**%Tiempo**	**%Calidad**	**%Tiempo**
Letter R.	20000	16	60	-1.07	86.10	-0.22	68.21	-5.43	88.58
Letter R.	20000	16	70	-0.95	86.38	-0.13	73.04	-5.47	88.74
Letter R.	20000	16	80	-0.94	86.94	-0.16	75.22	-5.32	90.24
Letter R.	20000	16	90	-0.80	83.92	-0.21	74.81	-5.30	88.08
Letter R.	20000	16	100	-0.72	83.23	-0.24	74.90	-5.34	87.69
New York	245057	3	100	-4.32	73.25	-1.44	78.98	-23.80	89.39
Skin	657308	2	100	-4.46	84.99	-0.86	75.15	-24.97	94.22
Uci-Sc	600	60	50	-0.05	47.22	-0.06	35.72	-2.18	52.50
Wind	6574	15	20	-0.58	87.16	-0.03	60.58	-2.84	89.22
Wind	6574	15	40	-0.56	83.50	-0.13	70.39	-3.12	88.57
Wind	6574	15	60	-0.39	79.03	-0.21	75.05	-3.03	88.59
Wind	6574	15	80	-0.28	72.05	-0.24	74.54	-2.73	85.95
Wind	6574	15	100	-0.19	67.26	-0.24	74.29	-2.68	83.63
Wind	6574	15	120	-0.15	60.63	-0.26	73.71	-2.62	82.13
Wind	6574	15	140	-0.10	57.76	-0.31	74.96	-2.59	81.64
Wind	6574	15	160	-0.10	56.01	-0.39	77.10	-2.67	82.56

Tabla B.4 Resultados experimentales con instancias sintéticas en términos de porcentaje de pérdida de calidad y porcentaje de reducción de tiempo

Instancia	***n***	***d***	***k***	Early Classification		Enhanced K-means		Pattern Reduction	
				%Calidad	**%Tiempo**	**%Calidad**	**%Tiempo**	**%Calidad**	**%Tiempo**
2500	2500	2	50	-1.60	80.42	-0.86	72.63	-5.61	78.89
2500	2500	2	100	-0.78	73.59	-1.20	76.85	-4.39	76.22
2500	2500	2	200	-0.17	54.94	-1.38	74.44	-3.24	65.28
2500	2500	2	400	-0.02	33.00	-1.13	70.50	-1.88	52.83
2500	2500	2	800	0.00	11.70	-0.51	65.13	-0.52	51.90
10000	10000	2	50	-2.65	92.80	-0.33	74.49	-7.50	88.74
10000	10000	2	100	-2.22	90.87	-0.68	79.13	-6.82	87.70
10000	10000	2	200	-1.35	84.45	-0.84	79.95	-5.51	82.96
10000	10000	2	400	-0.60	74.37	-1.13	80.58	-4.54	78.66
10000	10000	2	800	-0.09	53.51	-1.26	77.88	-3.13	68.70
20000	20000	2	50	-3.04	94.69	-0.28	73.63	-7.95	89.72
20000	20000	2	100	-2.60	94.19	-0.45	79.21	-7.37	89.99
20000	20000	2	200	-1.92	91.18	-0.56	81.48	-6.23	87.54
20000	20000	2	400	-1.13	86.03	-0.78	83.28	-5.29	84.89
20000	20000	2	800	-0.45	74.16	-1.08	81.89	-4.22	77.86
40000	40000	2	50	-2.67	95.67	-0.10	67.83	-6.74	89.47
40000	40000	2	100	-2.32	96.07	-0.20	77.15	-6.31	91.30
40000	40000	2	200	-2.18	94.69	-0.35	82.09	-6.57	90.63
40000	40000	2	400	-1.58	91.65	-0.49	83.40	-5.73	88.11
40000	40000	2	800	-1.01	85.62	-0.74	83.74	-5.23	84.16
DSH1	6000	2	50	-2.99	89.09	-1.01	74.40	-7.30	84.06

Tabla B.4 Resultados experimentales con instancias sintéticas en términos de porcentaje de pérdida de calidad y porcentaje de reducción de tiempo (continuación)

Instancia	***n***	***d***	***k***	**Early Classification**		**Enhanced K-means**		**Pattern Reduction**	
				%Calidad	**%Tiempo**	**%Calidad**	**%Tiempo**	**%Calidad**	**%Tiempo**
DSH2	6000	10	50	-0.57	82.19	-0.14	71.57	-2.59	86.19
DSH3	6000	25	50	-0.07	63.05	-0.09	68.97	-1.16	83.11
DSH4	6000	50	50	0.00	38.67	-0.04	66.28	-0.71	79.39
DSH5	6000	100	50	-2.10	85.36	-1.18	79.68	-6.84	82.93
DSH6	6000	250	50	0.00	7.82	-0.01	63.86	-0.17	72.59
DSH7	6000	500	50	0.00	3.44	-0.01	60.65	-0.07	66.11
DSH8	6000	1000	50	0.00	5.16	0.00	60.56	-0.02	63.37
DSL1	60000	2	50	-3.11	96.10	-0.08	66.72	-8.05	90.19
DSL2	600000	2	50	-3.07	97.82	-0.02	53.91	-7.43	90.07

Anexo

C

Pruebas estadísticas

En este Anexo se muestran los resultados obtenidos en las pruebas estadísticas. Dichas pruebas fueron realizadas con el software estadístico SPSS. Mediante el contraste de los supuestos paramétricos se determinó el uso de pruebas estadísticas no paramétricas.

C-1 Prueba de Friedman

La prueba de Friedman es una prueba no paramétrica que se utiliza como alternativa de la prueba ANOVA para medias repetidas, la cual, se basa en rangos de orden y se compara con valores críticos de una distribución Chi Cuadrada (χ^2).

Los pasos para el desarrollo de la prueba son:

1) Definir las hipótesis nula H_0 y alternativa H_1.

2) Ordenar los valores de las k muestras conjuntamente.

3) Asignar un rango de orden a cada valor.

4) Corregir las ligaduras o empates, esto es, si existen rangos diferentes asignados a mismos valores se obtiene la media, la cual será el valor del rango para dichos valores.

5) Obtener la suma de rangos de las k muestras.

6) Calcular el estadístico F_R mediante la expresión C-1.1

$$F_R = \left(\frac{12}{n\,k(k+1)}\right)\left(\sum_{j=1}^{k} R_j^2 - 3n(k+1)\right) \qquad \text{(C-1.1)}$$

Donde n indica el número de objetos por cada muestra, k es el número de muestras a comparar y R_j^2 es el cuadrado del total de los rangos para el grupo j.

7) Tomar la decisión: Rechazar H_0 si el estadístico $F_R$es mayor al valor crítico de la distribución Chi-Cuadrada con k-1 grupos y α nivel de significancia.

Es importante mencionar que este procedimiento se complica al aumentar el número de muestras a comparar y el tamaño de las mismas, para ello, existen paquetes estadísticos, tales como SPSS, que apoyan a la solución de esta prueba. Las muestras obtenidas en este estudio son muestras pareadas lo que justifica el uso de la prueba de *Friedman* con el objetivo de determinar si existen diferencias significativas entre las muestras. Por lo tanto, se establece α=0.05 (5% de error) y las hipótesis siguientes:

H_0: No existen diferencias significativas entre las muestras correspondientes a los resultados experimentales de SSE y tiempo obtenidas por la ejecución de 4 algoritmos.

H_1: Existen diferencias significativas entre las muestras correspondientes a los resultados experimentales de SSE y tiempo obtenidas por la ejecución de 4 algoritmos.

El resultado de la prueba de *Friedman* se muestra en la Tabla C.1, donde, se puede observar que el valor de significancia para las k-1 muestras (Gl=2) con α=0.05

es menor al 5%, por lo tanto, se rechaza la hipótesis nula y se determina que existe evidencia estadística para decir que un algoritmo es mejor que otro.

Tabla C.1 Prueba de Friedman de significancia estadística

n	61
Chi-cuadrado	91.7050
Gl	2
Significancia Asintótica	0

C-2 Pruebas de Wilcoxon

La prueba de Wilcoxon compara dos muestras pareadas o relacionadas respecto a sus medianas, con el fin de determinar diferencias significativas entre los grupos. Esta prueba se basa en la asignación de rangos de orden y se utiliza como alternativa no paramétrica de la prueba t-student para muestras pareadas que no cumplen con los supuestos de normalidad y homogeneidad.

Para la ejecución de la prueba de Wilcoxon se siguen los siguientes pasos:

1. Definir las hipótesis nula H_0 y alternativa H_1.
2. Calcular las diferencias entre las muestras a comparar.
   -Eliminar las diferencias nulas (0), ya que no aportan información para el estadístico.
3. Ordenar las diferencias prescindiendo de los signos.
4. Asignar un rango de orden.
5. Corregir las ligaduras o empates, esto es, si existen rangos diferentes asignados a mismos valores se obtiene la media, la cual será el valor del rango para dichos valores.
6. Obtener la suma de rangos tanto para las diferencias positivas (W_+) como negativas (w_-).
7. Definir el estadístico $W_s = min[w_+, w_-]$.

8. Tomar la decisión de aceptación o rechazo bajo las siguientes condiciones:

   -Si $n \leq 30$ contrastar según la tabla de Wilcoxon, donde el valor del estadístico se comprara con el valor crítico. Si el valor del estadístico es menor al valor crítico, entonces se rechaza la hipótesis nula

   -Si $n > 30$ contrastar con la aproximación de la normal.

Con el objetivo de determinar entre qué pares de muestras existe tal diferencia y validar nuestros resultados experimentales se aplicaron pruebas de Wilcoxon estableciendo que α=0.05 (5% de error) y las hipótesis siguientes:

H_0= Las diferencias en las soluciones de error al cuadrado obtenidas por la ejecución de dos algoritmos son debido a la aleatoriedad.
H_1= Las diferencias en las soluciones de error al cuadrado obtenidas por la ejecución de dos algoritmos no se deben a la aleatoriedad.

Mediante la expresión C-1.2 se calculó el número de posibles combinaciones, de modo que en la Tabla C.2 se presentan los resultados de múltiples pruebas de Wilcoxon donde n indica el número de elementos que aportan información para la prueba, W_s es el mínimo valor absoluto de la suma de los rangos positivos y negativos, y C_v corresponde al valor crítico de n. Cuando $W_s < C_v$ se rechaza H_0. En la Tabla C.2 se muestran en amarillo los casos en que se acepta H_0, de manera general se puede observar que para el 98% los casos $W_s < C_v$, con lo que se concluye que con un nivel de confianza del 95%, existe suficiente evidencia para apoyar que nuestros resultados no se obtienen debido a la aleatoriedad (H_1) y determinar si existen casos en que un algoritmo presenta mejores resultados que otro.

$$\frac{k * (k-1)}{2} \tag{C-1.2}$$

Donde k indica el número de muestras a comparar.

Tabla C.2 Resultados de las pruebas de Wilcoxon

Instancia	Early Classification vs K-means			Enhanced K-means vs K-means			Pattern Reduction vs K-means			Early Classification vs Enhanced K-means			Early Classification vs Pattern Reduction			Enhanced K-means vs Pattern Reduction		
	n	W_s	C_v	n	W_s	C_v	n	W_s	C_v	n	W_s	C_v	n	W_s	C_v	n	W_s	C_v
Abalone																		
200	29	0	141	30	0	152	30	0	152	30	15	152	30	0	152	30	0	152
400	15	0	30	30	0	152	30	0	152	30	1	152	30	0	152	30	0	152
600	5	0	1	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152
800	1	0.317	0.05	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152
1000	0	0	0	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152
Letters																		
40	30	0	152	30	120	152	30	0	152	30	0	152	30	0	152	30	0	152
50	30	0	152	30	141	152	30	0	152	30	0	152	30	0	152	30	0	152
60	30	0	152	30	81	152	30	0	152	30	0	152	30	0	152	30	0	152
70	30	0	152	30	95	152	30	0	152	30	0	152	30	0	152	30	0	152
80	30	0	152	30	84	152	30	0	152	30	0	152	30	0	152	30	0	152
90	30	0	152	30	62	152	30	0	152	30	0	152	30	0	152	30	0	152
100	30	0	152	30	45	152	30	0	152									
Wind																		
20	30	0	152	30	159	152	30	0	152	30	0	152	30	0	152	30	0	152
40	30	0	152	30	45	152	30	0	152	30	1	152	30	0	152	30	0	152
60	30	0	152	30	16	152	30	0	152	30	5	152	30	0	152	30	0	152
80	30	0	152	30	1	152	30	0	152	30	132	152	30	0	152	30	0	152
100	30	0	152	30	1	152	30	0	152	30	115	152	30	0	152	30	0	152
120	30	0	152	30	0	152	30	0	152	30	28	152	30	0	152	30	0	152
140	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152
160	30	0	152	30	2	152	30	0	152	30	0	152	30	0	152	30	0	152
2500																		
50	30	0	152	30	2	152	30	0	152	30	0	152	30	0	152	30	0	152
100	30	0	152	30	0	152	30	0	152	30	31	152	30	0	152	30	0	152
200	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152
400	11	0	14	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152
800	0	0	0	30	0	152	30	0	152	30	0	152	30	0	152	30	168.5	152
10000																		
50	30	0	152	30	6	152	30	0	152	30	0	152	30	0	152	30	0	152
100	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152
200	30	0	152	30	0	152	30	0	152	30	10	152	30	0	152	30	0	152
400	30	0	152	30	0	152	30	0	152	30	1	152	30	0	152	30	0	152

Tabla C.2 Resultados de las pruebas de Wilcoxon (continuación)

Instancia	Early Classification vs K-means			Enhanced K-means vs K-means			Pattern Reduction vs K-means			Early Classification vs Enhanced K-means			Early Classification vs Pattern Reduction			Enhanced K-means vs Pattern Reduction		
	n	W_s	C_v	n	W_s	C_v	n	W_s	C_v	n	W_s	C_v	n	W_s	C_v	n	W_s	C_v
800	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152
20000																		
50	30	0	152	30	17	152	30	0	152	30	0	152	30	0	152	30	0	152
100	30	0	152	30	1	152	30	0	152	30	0	152	30	0	152	30	0	152
200	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152
400	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152
800	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152
40000																		
50	30	0	152	30	71	152	30	0	152	30	0	152	30	0	152	30	0	152
100	30	0	152	30	19	152	30	0	152	30	0	152	30	0	152	30	0	152
200	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152
400	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152
800	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152	30	0	152
Iris																		
3	14	5	26	9	14	8	26	1	110	19	38.5	54	26	0	110	27	15	120
5	18	1	47	14	24.5	26	28	0	130	25	53	101	28	0	130	28	0	130
10	13	3	21	24	7.5	92	30	0	152	24	39.5	92	30	0	152	30	1	152
20	5	0	1	27	8.5	120	30	0	152	27	8.5	120	30	0	152	30	8.5	152
30	2	0.18	0.05	26	8	110	30	0	152	26	8	110	30	0	152	30	13	152
40	3	0.109	0.05	21	1	68	30	1	152	21	1	68	30	1	152	30	10	152
50	0	0	0	23	1	83	30	0	152	23	1	83	30	0	152	30	3	152
Concrete																		
100	9	0	8	30	1	152	30	0	152	30	1	152	30	0	152	30	0	152
Skin																		
100	30	0	152	30	56	152	30	0	152	30	0	152	30	0	152	30	0	152
New York																		
100	30	0	152	30	35	152	30	0	152	30	6	152	30	0	152	30	0	152
uci-sc																		
50	9	1	8	16	12	36	29	0	141	30	0	152	30	0	152	30	0	152
DSH1																		
50	30	0	152	30	4	152	30	0	152	30	0	152	30	0	152	30	0	152
DSH2																		
50	30	0	152	30	19	152	30	0	152	30	0	152	30	0	152	30	0	152
DSH3																		
50	30	1	152	30	20	152	30	0	152	30	153.5	152	30	0	152	30	0	152

Tabla C.2 Resultados de las pruebas de Wilcoxon (continuación)

Instancia	Early Classification vs K-means			Enhanced K-means vs K-means			Pattern Reduction vs K-means			Early Classification vs Enhanced K-means			Early Classification vs Pattern Reduction			Enhanced K-means vs Pattern Reduction		
	n	W_s	C_v	n	W_s	C_v	n	W_s	C_v	n	W_s	C_v	n	W_s	C_v	n	W_s	C_v
DSH4																		
50	16	63	36	30	1	152	30	0	152	30	2	152	30	0	152	30	0	152
DsH5																		
50	30	0	152	30	0	152	30	0	152	30	1	152	30	1	152	30	0	152
DSH6																		
50	0	0	0	30	5	152	30	0	152	30	5	152	30	0	152	30	0	152
DSH7																		
50	0	0	0	30	10	152	30	0	152	30	0	152	30	0	152	30	0	152
DSH8																		
50	0	0	0	30	29	152	30	0	152	30	29	152	30	0	152	30	0	152
DSL1																		
50	30	0	152	30	132.5	152	30	0	152	30	0	152	30	0	152	30	0	152
DSL2																		
50	30	0	152	30	182	152	30	0	152	30	0	152	30	0	152	30	0	152

cenidet

Centro Nacional de Investigación
y Desarrollo Tecnológico